

# **Problemas de programación lineal entera. Aplicación a un problema de planificación de turnos hospitalarios**



**Tomás Malón Melendo**  
Trabajo de fin de grado en Matemáticas  
Universidad de Zaragoza

Director del trabajo: Pedro M. Mateo Collazos  
26 de junio de 2020



# Summary

Integer-Linear Programming is an area of Operation Research which arises in practically every area of application of Mathematical Programming and allows us to model a wide range of real-life problems. In 1947, the American mathematician George B. Dantzig developed the *Simplex Method*, providing a standard and effective technique for solving Linear Programming Problems. Nonetheless, the computation of Integer-Linear Programming Problems is remarkably higher in comparison to the latter. Throughout the late 20th century, a handful of techniques have been proposed, aiming at approaching its resolution effectively.

The two paramount Integer-Linear Programming methods are: *Cutting-Plane Method* and *Branch-and-Bound Method*. The cutting-plane method was introduced by the American mathematician Ralph E. Gomory in 1958 and is characterized by its finiteness, however, it has currently become obsolete as the number of iterations – despite being finite – can happen to be enormous. Nowadays, branch-and-bound is one of the best-known methods and, what is more, is broadly implemented by the majority of state-of-the-art optimization software packages. As indicated in Cook [1], this algorithm was developed throughout 1950 with contributions from the mathematicians Harry M. Markowitz, Alan S. Manne, Willard L. Eastman, Ailsa H. Land and Alison G. Doig.

Despite the better performance of the latter method, it is nowadays more common the usage of a combination of both, which is called the *Branch-and-Cut Method*. It is based on the branch-and-bound algorithm, enabling the possibility of adding *cutting planes* to resolve the associated linear problems. This combination has proven to be more effective compared to the individual use of both methods. However, this method is not the subject of this paper but we uniquely underline its existence in this summary.

This paper aims at addressing in detail the Cutting-Plane Method and Branch-and-Bound Method, deepening the knowledge previously acquired in the module of Operation Research. Our main reference to develop these are the following books: Nemhauser & Wolsey [8, chapter II.4] and Murty [7, chapters 13, 14, 15]. Additionally, the aim of this paper is to present, formulate and solve a real-world application.

In the first chapter we will introduce the Integer-Linear Programming, defining its main characteristics, alongside enumerating the uppermost problems which can be modelled. Afterwards, as a brief introduction to the developed methods, the *Rounding off Method* is showed owing to be regarded as a practical and sensible algorithm, especially for someone who starts studying this breed of problems. Nevertheless, it will be indicated that its application is indeed limited to a narrow set of problems.

The second chapter develops the Cutting-Plane Method. We will firstly introduce the general formulation for both the *pure integer problem* and its *relaxed* counterpart, establishing the motivation of the method. Subsequently, once the cutting planes are defined, we will focus on the *Lexicographic Dual Simplex Algorithm*, which will be indeed the basis of the underlying theory of the method. This algorithm is a modification of the well-known *Dual Simplex Algorithm* and differs in the selection criteria of the non-basic variables entering and leaving the basis, enabling us to prove the absence of cycling and its finiteness. Besides, in cutting plane literature, the Lexicographic Dual Simplex tableaus are recorded differently as we are accustomed to – in any case, they are basically a transposition of the tabular tableaus used in the Simplex and Dual Simplex Methods. Then, the *Gomory Fractional Cutting-Plane Algorithm* is introduced, deriving the *Gomory fractional cut* and, we will explain how the cut is added to the optimum tableau – the original presentation of the method can be found in Gomory [4]. Eventually, we will study a set of propositions which will enable us to prove the theorem of finite convergence of

the method, which was presented in Gomory [2]. Additionally, we will extend Gomory fractional cuts to Mixed-Integer Programming Problems. In this extension, the finiteness will be preserved and the proof can be found in Gomory [3].

The third chapter is devoted to the Branch-and-Bound Method. As in the previous chapter, we will firstly introduce the motivation for the development of the method. Subsequently, we will present the concept of *divide and conquer* – which is indeed the basis of the method – and will allow us to find an optimal solution without having to consider the set of all the feasible solutions. Afterwards, the *Branch-and-Bound Algorithm* is presented indicating its characteristics. A remarkable point of the algorithm will be the *division* concept, which will actually provide us with a tool to partition the initial feasible region. Eventually, four of the most comprehensive strategies for both branching variable and node selections will be briefly discussed – both play an outstanding role in the algorithm efficiency.

To illustrate the practical resolution of integer linear problems, Appendix A proposes a maximum problem which we will solve with the two methods developed.

On the other hand, this paper addresses the *Emergency Room Physician Scheduling Problem*, which falls in the category of *NP-hard* problems owing to the consideration of a large number of restrictions, the great numbers of physicians contemplated and the planning horizon. This paper solves the physician scheduling problem for 43 physicians over a planning horizon corresponding to the year 2020, implementing the model in the optimization software package CPLEX. The problem will be modelled as a Mixed-Integer Programming Problem. The set of constraints to be considered ought to be classified into four categories: (a) working hours constraints, (b) demand covering constraints, (c) ergonomic constraints and (d) fairness constraints. Furthermore, all these constraints can be treated as: (1) *hard constraints* (a, b, c), which shall be satisfied for the schedule to be valid and are imposed by-law, (2) *soft restrictions* (d), which can be violated with a penalty in the objective function.

The physician scheduling problem shares many characteristics with the *Nurse Scheduling Problem*, which has been actually developed in a large number of research papers – unlike our problem. Although at a glance, their resolutions seem likely to be analogous, it happens to be quite the opposite on account of a handful of differences which makes the physician scheduling problem far more complex to solve. Some of these differences are as follow: (1) the number of types of shifts, in the nurse case is considerably lower, (2) the objective function, the nurse problem is subject to cost-related goals, whereas the physician problem's objective function is driven by fairness constraints.

In the fourth and last chapter, we will illustrate the general formulation of the physician scheduling problem, defining the parameters and decision variables regarded. Moreover, we will indicate that the solutions obtained guarantee the fairness of the schedule. As an extension to this fourth chapter, Appendix B is presented to show in detail the notation, data and mathematical model of the problem – deriving the expressions of each of the constraints. Eventually, the CPLEX code built alongside the results and the schedule obtained for the month of January 2020 will be presented.

Finally, to complete this paper, we will present the conclusions, briefly reviewing the principle points addressed and, we will underline the current extension of the physician scheduling problem in ongoing research areas.

# Resumen

La Programación Lineal Entera es un área de la Investigación Operativa que surge en prácticamente todas las áreas de aplicación de la Programación Matemática y permite modelar una gran variedad de problemas del mundo real. En 1947, el matemático estadounidense George B. Dantzig desarrolló el *Método del Simplex* aportando una herramienta muy eficaz para resolver problemas de Programación Lineal, manteniéndose en la actualidad como el método más extendido para resolver esta clase de problemas. Sin embargo, la resolución de los problemas de Programación Lineal Entera es notablemente mayor en comparación a estos últimos. A lo largo de finales del siglo XX, se han propuestos varios métodos para abordar su resolución de una forma eficiente.

Los dos métodos más famosos de la Programación Lineal Entera son: *Método de planos de corte* y, *Método de ramificación y acotación*. El método de planos de corte fue desarrollado por el matemático estadounidense Ralph E. Gomory en 1958; se caracteriza por la garantía de su convergencia finita, sin embargo hoy en día se ha quedado obsoleto debido a que el número de iteraciones – a pesar de ser finito – puede ser muy grande. En la actualidad, uno de los métodos más populares es el método de ramificación y acotación y, el que la mayoría de los softwares de optimización tienen implementado. Como se indica en Cook [1], este método fue desarrollado a lo largo de 1950 con las contribuciones de los matemáticos Harry M. Markowitz, Alan S. Manne, Willard L. Eastman, Ailsa H. Land y Alison G. Doig.

A pesar de la mayor eficiencia de este último método, actualmente es más habitual el empleo de una combinación de ambos, que recibe el nombre de *Método de ramificación y corte*, se basa en el método de ramificación y acotación, añadiéndole la posibilidad de generar *planos de corte* para resolver los problemas relajados; esta combinación ha probado ser más eficiente que el uso individual de ambos métodos. Este método no es objeto de este trabajo y únicamente se aprovecha este resumen para mencionar su existencia.

El objetivo de este trabajo es de desarrollar en detalle el Método de planos de corte y, el Método de ramificación y acotación, profundizando en los conocimientos adquiridos en esta materia en la asignatura de Investigación Operativa. Nuestra referencia principal para desarrollar los dos métodos son los libros: Nemhauser & Wolsey [8, capítulo II.4] y Murty [7, capítulos 13, 14, 15]. Adicionalmente, este trabajo tiene como objetivo presentar, formular y resolver un problema práctico.

En el primer capítulo mostraremos una introducción a la Programación Lineal Entera, definiendo sus características básicas y, enumerando los problemas más destacados que se permiten modelizar. Posteriormente, como introducción a los métodos que se desarrollan, se muestra los *Métodos de redondeo* – que resultan un método sensato de desarrollar cuando se pasa por primera vez de la programación lineal a la programación lineal entera, pero se observará que su aplicación está limitada a un número reducido de problemas.

En el segundo capítulo se desarrolla el Método de planos de corte. En la introducción mostraremos la formulación general del *problema lineal entero puro* y del *problema relajado* asociado, estableciendo la motivación del método. Seguidamente, tras definir los planos de corte, vamos a presentar el *Algoritmo del Simplex dual lexicográfico* que será la base del desarrollo del método. Este algoritmo es una modificación del conocido *Algoritmo del Simplex dual* que se diferencia en los criterios de elección de las variables de entrada y salida en la base empleados, permitiéndonos probar la ausencia de ciclos y la finitud del algoritmo. Notar que las tablas con la que se trabajarán en este algoritmo son una transposición de las tablas que se suelen emplear en los algoritmos del Simplex primal y Simplex dual. Posteriormente,

se presenta el *Algoritmo fraccional de Gomory*, derivando el *corte fraccional de Gomory* y mostraremos cómo se añade el corte a la tabla óptima del problema correspondiente – la presentación original del método se puede encontrar en Gomory [4]. Finalmente, estudiaremos una serie de proposiciones que nos permitirá probar el teorema de la convergencia finita del método, el cual se presentó en Gomory [2]. Adicionalmente, extenderemos el corte de Gomory para los Problemas Lineales Enteros Mixtos. En esta extensión, la finitud del método se conservará y se puede encontrar su demostración en Gomory [3].

El tercer capítulo está dedicado al Método de ramificación y acotación. Al igual que en el anterior capítulo, primero mostraremos la motivación para el desarrollo del método. A continuación, presentaremos el concepto de *divide y vencerás* que es la base de este método y, nos permitirá encontrar una solución óptima sin tener que considerar el conjunto de todas las soluciones factibles. Después, se presenta el *Algoritmo de ramificación y acotación* indicando sus características. Un punto importante del algoritmo será el concepto de *ramificación* que nos proporcionará la herramienta para realizar las particiones de la región de factibilidad inicial. Para finalizar, se muestran cuatro de los criterios más habituales para la selección de la variable para ramificar y selección de los subproblemas – ambos clave en la eficiencia del algoritmo.

Para ilustrar la resolución práctica de los problemas lineales enteros, en el Anexo A se propone un problema de máximo que será resuelto con los dos métodos desarrollados.

En este trabajo se presenta el *Problema de planificación de turnos hospitalarios en un Servicio de Urgencias*. Se trata de un problema que entra dentro de la categoría de complejidad computacional *NP-complejo* debido a la consideración de una gran número de restricciones, el elevado número de médicos que se contempla y el plano temporal en la que se realiza la planificación. En este trabajo se resuelve la planificación de una plantilla de 43 médicos para el año 2020 mediante el lenguaje de modelado CPLEX. Este problema se modelizará como un problema de Programación Lineal Entero Binario Mixto. El conjunto de restricciones que se considerará se clasifican en: (a) disponibilidad horaria, (b) cubrimiento, (c) ergonomía y (d) equidad. Las cuales se agrupan en: (1) *restricciones fuertes* (a, b, c), se deben de cumplir de manera obligatoria por Ley y, (2) *restricciones suaves* (d), no son de obligatoriedad en su cumplimiento pero sí son deseables.

Este problema muestra una estructura similar con el *Problema de planificación de turnos de enfermería*, el cual se ha desarrollado un gran número de investigaciones en la literatura – a diferencia del problema en el que se centra el trabajo. Aunque a priori pueda parecer que la resolución de ambos es análoga, cabe remarcar que existen una serie de diferencias que hacen que el problema de turnos hospitalarios sea más complejo de resolver. Algunas de estas diferencias son: (1) el número de tipos de turnos, en el caso de enfermería es considerablemente menor, (2) la función objetivo, para el problema de los turnos de enfermería está sujeta a términos de costes económicos, mientras que la función objetivo para los turnos hospitalarios está sujeta a satisfacer restricciones de equidad.

En el cuarto y último capítulo, vamos a describir el problema de planificación de turnos hospitalarios de forma general, definiendo los parámetros y las variables de decisión empleadas. Por último se muestra la soluciones obtenidas, donde comprobaremos que efectivamente la planificación obtenida es justa. Como extensión a este cuarto capítulo, se presenta el Anexo B para mostrar en detalle la notación, los datos y la formulación matemática del problema – derivando las expresiones para cada una de las restricciones. Adicionalmente, se mostrará el código de resolución en CPLEX, los resultados y la planificación obtenida para el mes de Enero de 2020.

Finalmente, para concluir este trabajo, incluimos un apartado de conclusiones donde brevemente repasaremos los puntos que hemos estudiado y, mostraremos la extensión que tiene en la actualidad el problema de planificación abordado.

# Índice general

<b>Summary</b>	<b>iii</b>
<b>Resumen</b>	<b>v</b>
<b>1. Introducción a la programación lineal entera</b>	<b>1</b>
<b>2. Método de planos de corte</b>	<b>3</b>
2.1. Introducción . . . . .	3
2.2. Planos de corte . . . . .	3
2.3. Algoritmo del Simplex dual lexicográfico . . . . .	4
2.4. Método fraccional de Gomory para problemas lineales enteros puros . . . . .	7
2.4.1. Corte fraccional de Gomory . . . . .	8
2.4.2. Convergencia finita del método fraccional de Gomory . . . . .	10
2.5. Corte fraccional de Gomory para problemas lineales enteros mixtos . . . . .	14
<b>3. Método de ramificación y acotación</b>	<b>17</b>
3.1. Introducción . . . . .	17
3.2. Algoritmo de ramificación y acotación . . . . .	17
3.3. Ramificación . . . . .	19
3.4. Criterios de selección para el algoritmo de ramificación y acotación . . . . .	22
3.4.1. Criterio de selección de la variable para ramificar . . . . .	22
3.4.2. Criterio de selección de los subproblemas . . . . .	22
<b>4. Aplicación a un problema de planificación de turnos hospitalarios</b>	<b>23</b>
<b>Conclusiones</b>	<b>25</b>
<b>Bibliografía</b>	<b>27</b>
<b>Anexo A. Ejemplos de resolución aplicando los métodos propuestos</b>	<b>29</b>
A.1. Resolución con el método fraccional de Gomory . . . . .	29
A.2. Resolución con el método de ramificación y acotación . . . . .	33
<b>Anexo B. Problema de planificación de turnos hospitalarios</b>	<b>41</b>
B.1. Notación . . . . .	41
B.2. Datos . . . . .	42
B.3. Formulación matemática . . . . .	44
B.4. Código de resolución en CPLEX . . . . .	49
B.5. Resultados . . . . .	60





# Capítulo 1

## Introducción a la programación lineal entera

Los inicios de la Investigación Operativa (IO) como disciplina se remontan con el comienzo de la Segunda Guerra Mundial para abordar el estudio de problemas estratégicos y tácticos debido a la limitación de recursos. Una vez terminada, generó gran interés la posibilidad de su aplicación al ámbito civil debido al gran éxito que tuvo en el campo militar. El importante desarrollo de la IO durante esta época permitió – como consecuencia – entre otros, su aplicación directa al sector industrial debido al aumento de la complejidad y la especialización del tejido empresarial. En la actualidad, la IO se aplica en un amplio abanico de áreas las cuales se encuentran: transporte, telecomunicaciones, sector financiero, sector de la salud y servicios públicos, permitiendo una enorme mejora en la eficiencia y productividad de las actividades de todos estos sectores.

Un área muy destaca dentro de la Investigación Operativa es la *Programación Lineal (PL)*, considerada como uno de los avances más importantes de mediados del siglo XX. El desarrollo del *Método Simplex* por George B. Dantzig en 1947, resulta un procedimiento muy eficiente en la resolución de estos problemas lineales, incluso los de gran tamaño. Esto permite que en la actualidad la PL sea considerada como una herramienta básica empleada por empresas para organizar de forma eficiente y productiva su funcionamiento. Sin embargo, en el mundo real se encuentran muchos problemas donde las variables no tienen sentido considerarlas continuas, a diferencia de la asunción realizada en los problemas de PL sobre las variables de decisión. Es por ello, la necesidad de crear una teoría que permita una resolución eficiente de este tipo de problemas, donde las variables tienen que tomar valores enteros.

Recibe el nombre de *Programación Lineal Entera (PLE)* a todos aquellos problemas de optimización lineal cuyas variables no están obligadas a tomar valores reales. De esta forma, todos estos problemas se pueden formular como problemas de PL añadiendo restricciones adicionales sobre la imposición de que algunas o todas las variables tomen valores enteros. De esta manera, se encuentran los siguientes tipos de problemas de PLE: (1) *Problemas lineales enteros puros (PLE puros)*, en los cuales todas las variables del problema están restringidas a tomar valores enteros y, (2) *Problemas lineales enteros mixtos (PLEM)*, en donde algunas variables son continuas y otras se restringen a tomar valores enteros. Dentro de cada uno de estos problemas, encontramos las siguientes subclases: (a) problemas de PLE donde las variables de decisión están restringidas a tomar los valores 0 ó 1, *variables binarias* y, (b) problemas de PLE donde las variables toman valores enteros no negativos, *variables generales*.

Los problemas de PLE permiten modelar muchas más situaciones que los problemas de PL debido al hecho de que en muchos problemas del mundo real, activos y recursos como pueden ser: maquinaria, mercancías, paquetes – entre otros – son indivisibles. Además, se encuentran otros problemas en los que se incluyen decisiones del tipo: *sí o no*, que son modeladas mediante el uso de variables binarias y reciben el nombre de *Problemas lineales enteros binarios (PLEB)*. Esta gran aplicabilidad e importancia del uso de la PLE, se pone de manifiesto en el caso de la aerolínea *Air New Zealand*, la cual se estima que consiguió ahorrar alrededor de 6.7 millones de dólares al año debido a la implementación de modelos de PLEB para optimizar la programación de su tripulación (Hillier y Lieberman [5], pp. 4,

435]). Otro ejemplo, lo podemos encontrar en la empresa *Waste Management, Inc.* (proveedor líder de servicios de manejo de basura en Estados Unidos), la cual desarrolló un modelo de PLEB mixto para la administración de rutas en los procesos de recolección y disposición de la basura, que le permitió obtener unos ahorros de 100 millones de dólares anuales (Hillier et al. [5, pp. 4, 466]).

A continuación enumeramos una serie de problemas destacados que se pueden formular como problemas lineales enteros: (1) *Problemas de carga fija*, (2) *Problemas de ubicación de plantas y problemas de transporte*, son dos problemas clásicos dentro de la literatura. El primero consiste en determinar la ubicación óptima de un conjunto de plantas (o almacenes) y las rutas de transporte que se abren. El segundo problema consiste en determinar la distribución de las mercancías – ambos problemas están sujetos a satisfacer unas demandas y minimizar unos costos dados. (3) *Problemas de selección de proyectos*, (4) *Problema de la mochila*, consiste en introducir un conjunto de artículos (cada uno con un peso y un valor asociado) a una mochila sujeto a que no exceda la capacidad de esta y, maximice su valor. Este problema es significativo ya que se ha probado que todo PLE puro puede formularse como un problema de este tipo. (5) *Problemas de optimización combinatoria*, se deben de establecer una combinación óptima de variables dentro de un conjunto de posibles combinaciones. (6) *Problemas de representación de conjuntos*, consisten en encontrar el representante<sup>1</sup> de menor cardinalidad que minimice unos costes dados – dentro de este tipo destacan *Problemas de localización de almacenes y, Problemas de reparto y rutas*. (7) *Problema del cartero chino*, consiste en encontrar la mínima ruta que recorra todos los nodos de un grafo, este tipo de problemas se aplica en la determinación de rutas óptimas de metros, sistemas de recogida de basura, entre otros. (8) *Problema del viajante*, un comerciante tiene que visitar  $n$  ciudades (con un costo asociado al desplazamiento a cada ciudad), se trata de determinar la ruta que el comerciante debe de recorrer para visitar todas las ciudades una única vez con el objetivo de minimizar los costos, sabiendo que tiene que empezar y terminar en la misma ciudad. (9) *Problemas de diseño de horarios*, en particular cabe destacar el problema de *Planificación de horarios de los turnos hospitalarios en un Servicio de Urgencias* que se presenta en el Capítulo 4.

A pesar de la amplia gama de problemas que el uso de la PLE permite modelar, la complejidad en la resolución de este tipo de problemas es mucho mayor que la de problemas de PL – aunque a priori pueda padecer lo contrario – ya que el número de soluciones factibles es mucho menor al imponer las condiciones de integridad. Este incremento en la complejidad se debe al hecho de que se pierde la garantía de que la solución óptima se encuentra en un *punto extremo*, siendo esto la clave de la enorme eficiencia del método Simplex. Es por ello, que la mayoría de los algoritmos empleados en la resolución de PLE se basa en la relajación de las condiciones de integridad y, el uso del algoritmo del Simplex. Por otra parte, a pesar de que los problemas de PLE puros con región de factibilidad acotada tengan un número finito de soluciones factibles, este número puede ser enorme. Una solución sencilla sería resolver el problema relajado aplicando el algoritmo Simplex y, después redondear los valores no enteros a enteros. Estos *Métodos de redondeo* pueden ser un método muy práctico especialmente cuando se conoce que las variables de decisión tienen que tomar valores muy grandes, de forma que el redondeo produce un error muy pequeño. Sin embargo, es frecuente que en muchos problemas nos encontremos con que una solución óptima del PL no es necesariamente factible para el PLE después de redondearla y, por tanto, es difícil decidir en qué sentido redondear para conservar la factibilidad – además del hecho de que la solución óptima no tiene porqué encontrarse necesariamente en alguno de los puntos vecinos de la solución óptima del PL.

El *Método de planos de corte* y, el *Método de ramificación y acotación* forman parte de los conocidos *Algoritmos de relajación* que consisten en resolver una secuencia de problemas de PL mediante el uso del algoritmo Simplex – tanto *Simplex primal* como *Simplex dual* – refinando la relajación en cada iteración, mediante la introducción de restricciones hasta que se consigue obtener la solución óptima del problema lineal entero. En los siguientes dos capítulos desarrollamos estos dos métodos.

<sup>1</sup>Dados  $n, m \in \mathbb{N}$ , sea el conjunto  $A = \{1, \dots, n\}$  y sea  $F = \{F_1, \dots, F_m\}$  una familia de subconjuntos de  $A$ , entonces un subconjunto  $E \subseteq A$  se dice que es un representante de  $F$  si  $E \cap F_i \neq \emptyset, \forall i = 1, \dots, m$ .

## Capítulo 2

# Método de planos de corte

### 2.1. Introducción

Sea el problema lineal entero puro en forma estándar

$$\begin{aligned} \text{Máx} \quad & z(x) = c^T x \\ \text{s.a:} \quad & Ax = b \\ & x \in \mathbb{Z}_+^n \end{aligned} \tag{PLE}$$

donde  $A \in \mathbb{Z}^{m \times n}$  de rango  $m$ ,  $c \in \mathbb{Z}^n$ ,  $b \in \mathbb{Z}^m$ ,  $m < n$ . Consideramos su problema relajado asociado

$$\begin{aligned} \text{Máx} \quad & z(x) = c^T x \\ \text{s.a:} \quad & Ax = b \\ & x \in \mathbb{R}_+^n \end{aligned} \tag{PLE_R}$$

Suponer que la región de factibilidad de  $(PLE_R)$  está acotada, de manera que existe un  $d \in \mathbb{Z}^+$  tal que

$$\{x \in \mathbb{R}_+^n : Ax = b\} \subseteq \{x \in \mathbb{R}_+^n : \sum_{j=1}^n x_j \leq d\}. \tag{2.1}$$

Resolvemos el problema relajado  $(PLE_R)$  aplicando el algoritmo del Simplex primal o Simplex dual. Si  $(PLE_R)$  es no factible, entonces  $(PLE)$  es también no factible. En caso contrario, sea  $\bar{x}$  una solución óptima de  $(PLE_R)$ . Si  $\bar{x} \in \mathbb{Z}^n$ , entonces  $\bar{x}$  es una solución óptima de  $(PLE)$ . En caso contrario, existen algunos puntos  $\bar{x}_j$  tales que  $\bar{x}_j \notin \mathbb{Z}$  y, por tanto,  $\bar{x}$  es una solución no factible de  $(PLE)$ .

En este capítulo, vamos a mostrar cómo podemos resolver el problema  $(PLE)$  empleando el *Método fraccional de Gomory*. La principal característica de este método, que probaremos al final del capítulo, es su convergencia en un número finito de pasos.

### 2.2. Planos de corte

En este apartado presentamos las definiciones de *plano de corte* y *corte* que se utilizarán a lo largo del capítulo.

**Definición 2.1.** Un *hiperplano*  $H \subseteq \mathbb{R}^n$  es una colección de puntos de la forma

$$H = \{x \in \mathbb{R}^n : p^T x = \alpha\},$$

donde  $p \in \mathbb{R}^n \setminus \{0\}$  y  $\alpha \in \mathbb{R}$ .

Un hiperplano  $H$  define dos semiespacios cerrados

$$H^+ = \{x \in \mathbb{R}^n : p^T x \geq \alpha\} \quad \text{y} \quad H^- = \{x \in \mathbb{R}^n : p^T x \leq \alpha\}.$$

Sea  $\bar{x}$  una solución óptima de  $(PLE_R)$  tal que  $\bar{x} \notin \mathbb{Z}^n$ .

**Definición 2.2.** Un *plano de corte* es un hiperplano  $H$  que separa estrictamente  $\bar{x}$  del conjunto de soluciones factibles enteras del problema relajado  $(PLE_R)$ . De esta manera, un hiperplano  $H$  es un *plano de corte* si verifica que  $\bar{x} \notin H^+$  y,  $x \in H^+$  para toda solución factible entera  $x$  de  $(PLE_R)$ . Por convenio se toma la caracterización con  $H^+$ , sin embargo la definición se mantiene válida tomando  $H^-$ .

**Definición 2.3.** Se llama *corte* al semiespacio cerrado positivo  $H^+$  de un plano de corte  $H$ .

### 2.3. Algoritmo del Simplex dual lexicográfico

Presentamos el algoritmo del *Simplex dual lexicográfico* que será clave para demostrar posteriormente la finitud del método fraccional de Gomory. Empezamos primero definiendo la relación de orden total  $(\overset{L}{\leq})$  sobre  $\mathbb{R}^n$ , conocida como *orden lexicográfico*.

**Definición 2.4.** Dados  $x, y \in \mathbb{R}^n$ , las relaciones de orden total  $(\overset{L}{\leq})$  y orden estricto  $(\overset{L}{<})$  sobre  $\mathbb{R}^n$  están definidas de la siguiente forma:

$$x \overset{L}{\leq} y \iff x \overset{L}{<} y \quad \vee \quad x = y,$$

y

$$x \overset{L}{<} y \iff \exists k \ni x_k < y_k \wedge x_i = y_i \quad \forall i < k.$$

Considerar una solución básica primal factible dual (SBPFD en adelante)  $(z, x)$  del problema  $(PLE_R)$  expresada en la forma:

$$\begin{aligned} \text{Máx} \quad z &= \bar{z} + \sum_{j \in \Gamma} \bar{c}_j(-x_j) \\ \text{s.a:} \quad x_i &= \bar{b}_i + \sum_{j \in \Gamma} \bar{a}_{i,j}(-x_j) \quad \forall i = 1, \dots, n \\ x_i &\in \mathbb{R}^+ \quad \forall i = 1, \dots, n \end{aligned} \tag{2.2}$$

asumiendo que las variable básicas son  $x_1, \dots, x_m$  y el resto son no básicas. Denotamos por  $\Gamma = \{m+1, \dots, n\}$  al conjunto de los índices de las variables no básicas. Así pues, para todo  $i \in \Gamma$  tenemos las igualdades  $x_i = x_i$ , ya que  $b_i = 0$ ,  $\bar{a}_{i,i} = -1$  y  $\bar{a}_{i,j} = 0 \quad \forall j \in \Gamma \setminus \{i\}$ .

Dentro de la literatura de planos de corte, el conjunto de las restricciones de (2.2) incluyendo la función objetivo se presentan en la siguiente tabla.

Variables	Términos constantes	$-x_{m+1}$	$\dots$	$-x_{m+j}$	$\dots$	$-x_n$
$z$	$\bar{z}$	$\bar{c}_{m+1}$	$\dots$	$\bar{c}_{m+j}$	$\dots$	$\bar{c}_n$
$x_1$	$\bar{b}_1$	$\bar{a}_{1,m+1}$	$\dots$	$\bar{a}_{1,m+j}$	$\dots$	$\bar{a}_{1,n}$
$x_2$	$\bar{b}_2$	$\bar{a}_{2,m+1}$	$\dots$	$\bar{a}_{2,m+j}$	$\dots$	$\bar{a}_{2,n}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
$x_i$	$\bar{b}_i$	$\bar{a}_{i,m+1}$	$\dots$	$\bar{a}_{i,m+j}$	$\dots$	$\bar{a}_{i,n}$
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
$x_m$	$\bar{b}_m$	$\bar{a}_{m,m+1}$	$\dots$	$\bar{a}_{m,m+j}$	$\dots$	$\bar{a}_{m,n}$
$x_{m+1}$	0	-1	$\dots$	0	$\dots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
$x_{m+j}$	0	0	$\dots$	-1	$\dots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
$x_n$	0	0	$\dots$	0	$\dots$	-1

Tabla 2.1: Representación de la SBPFD  $(z, x)$  de  $(PLE_R)$ .

Denotamos por  $(z, x)^T$  y  $D_0$  a las columnas *Variables* y *Términos constantes* de la Tabla 2.1 y, por  $D_j$  a la columna correspondiente a la variable  $-x_j$ , para todo  $j \in \Gamma$ . De este modo, la Tabla 2.1 se puede reescribir como

<i>Variables</i>	<i>Términos constantes</i>	$-x_{m+1}$	$\dots$	$-x_{m+j}$	$\dots$	$-x_n$
$(z, x)^T$	$D_0$	$D_{m+1}$	$\dots$	$D_{m+j}$	$\dots$	$D_n$

Tabla 2.2: Representación de la Tabla 2.1 por columnas.

Notar que al asumir que la región de factibilidad de  $(PLE_R)$  está acotada, en particular, por (2.1) se verifica

$$\sum_{j \in \Gamma} x_j \leq \sum_{j=1}^n x_j \leq d. \quad (2.3)$$

Por tanto, siempre podemos encontrar una SBPFD para poder iniciar el algoritmo simplex dual lexicográfico ya que por (2.3), basta con añadir la restricción redundante<sup>1</sup>

$$\beta + \sum_{j \in \Gamma} x_j = d, \quad \beta \geq 0$$

como la primera restricción de (2.2). Después, realizamos un cambio de base seleccionando  $\beta$  como variable de salida y  $x_k$  como variable de entrada, para un  $k \in \Gamma$  dado por:

$$\bar{c}_k = \min_{j \in \Gamma} \{\bar{c}_j\}. \quad (2.4)$$

De este modo, en la siguiente tabla habremos obtenido una SBPFD verificando  $\bar{c}_j \geq 0$ , para todo  $j \in \Gamma$ . Además, como  $(1, \bar{a}_{1,j}, \dots, \bar{a}_{n,j})^T \stackrel{L}{>} 0$ , se verificará que  $(\bar{c}_j, 1, \bar{a}_{1,j}, \dots, \bar{a}_{n,j})^T \stackrel{L}{>} 0 \quad \forall j \in \Gamma$ . Por tanto, podemos suponer, sin pérdida de generalidad, que siempre podemos empezar con una SBPFD  $(z, x)$  verificando  $D_j \stackrel{L}{>} 0 \quad \forall j \in \Gamma$ .

**Proposición 2.5.** Sea  $(z, x)$  una SBPFD de  $(PLE_R)$  tal que  $D_j \stackrel{L}{>} 0 \quad \forall j \in \Gamma$ . Si  $\bar{b}_i \geq 0 \quad \forall i = 1, \dots, n$ , entonces  $(z, x)$  es solución óptima y es la mayor solución factible lexicográfica de  $(PLE_R)$ .

*Demostración.* Como  $(z, x)^T = D_0 = (\bar{z}, \bar{b}_1, \dots, \bar{b}_m, 0, \dots, 0)^T$  es una SBPFD y, por hipótesis,  $\bar{b}_i \geq 0 \quad \forall i = 1, \dots, n$ , entonces se sigue que  $(z, x)$  es factible primal, luego es óptima. Además, es la mayor solución factible lexicográfica ya que por (2.2), cualquier otra solución factible de  $(PLE_R)$  es de la forma  $D_0 + \sum_{j \in \Gamma} D_j(-x_j)$  con  $D_j \stackrel{L}{>} 0, x_j \geq 0, \forall j \in \Gamma$ . Por tanto, se tiene que

$$D_0 \geq D_0 + \sum_{j \in \Gamma} D_j(-x_j),$$

probando que  $(z, x)$  es la mayor solución factible lexicográfica de  $(PLE_R)$ . □

El algoritmo simplex dual lexicográfico es una modificación del algoritmo simplex dual habitual en el que se establece una manera especial de determinar las variables salientes y entrantes. Estos nuevos procedimientos de selección se mostrarán en las proposiciones 2.6 y 2.7. Para un  $p \geq 0$ , suponer que hemos realizado la  $p$ -ésima iteración del algoritmo y, que la tabla resultante es no factible primal. Para  $p = 0$ , nos referiremos a que no hemos realizado ninguna iteración del algoritmo y, que la tabla resultante es la tabla inicial del problema.

<sup>1</sup>Método de la restricción artificial, equivalente dual al método de la Gran M.

**Proposición 2.6. Elección de la variable que abandona la base.** Sea  $(z, x)^T = D_0^p + \sum_{j \in \Gamma^p} D_j^p (-x_j^p)$  una SBPFD tal que  $D_j^p \stackrel{L}{>} 0 \quad \forall j \in \Gamma^p$  y,  $\bar{b}_i^p < 0$  para un  $i \in \{1, \dots, m\}$ . Al realizar una operación de cambio de base tomando  $x_i$  como variable de salida, entonces obtenemos que  $D_0^{p+1} \stackrel{L}{<} D_0^p$ . El criterio de selección de la variable que abandona la base es el siguiente, sale  $x_i$  en la que se alcanza el mínimo siguiente:

$$\bar{b}_i^p = \min_{1 \leq i \leq m} \{\bar{b}_i^p : \bar{b}_i^p < 0\}.$$

*Demostración.* Sea  $x_i$  la variable de salida. Para un  $k \in \Gamma^p$ , suponer que elegimos  $x_k$  como variable de entrada en la base, entonces como  $\bar{a}_{i,k}^p < 0$  (en la proposición 2.7 se muestra cómo elegir la variable de entrada  $x_k$  tal que  $\bar{a}_{i,k}^p < 0$ ) y, realizando operaciones de pivotaje sobre la columna  $D_0^p$ , se obtiene en la siguiente tabla que

$$D_0^{p+1} = D_0^p - \left(\frac{\bar{b}_i^p}{\bar{a}_{i,k}^p}\right) D_k^p \stackrel{L}{<} D_0^p,$$

ya que  $\bar{b}_i^p < 0$ ,  $\bar{a}_{i,k}^p < 0$  y,  $D_k^p \stackrel{L}{>} 0$  por hipótesis al tomar  $k \in \Gamma^p$ . □

Ahora mostramos el criterio de elección de la variable de entrada en la base, de manera que se verifique  $D_j^{p+1} \stackrel{L}{>} 0 \quad \forall j \in \Gamma^{p+1}$ . Así pues, la tabla resultante preservará la factibilidad dual.

**Proposición 2.7. Elección de la variable que entra en la base.** Suponer que  $D_j^p \stackrel{L}{>} 0 \quad \forall j \in \Gamma^p$ . Sea  $x_i$  la variable básica que abandona la base tal que  $\bar{b}_i^p < 0$ . Definimos el conjunto  $\Gamma_i^p = \{j \in \Gamma^p : \bar{a}_{i,j}^p < 0\}$ . Si  $\Gamma_i^p = \emptyset$ , entonces el problema  $(PLE_R)$  es no factible. En caso contrario, elegimos el índice  $k \in \Gamma_i^p$  que verifique

$$\frac{1}{\bar{a}_{i,k}^p} D_k^p \stackrel{L}{>} \frac{1}{\bar{a}_{i,j}^p} D_j^p, \quad \forall j \in \Gamma_i^p \setminus \{k\}, \quad (2.5)$$

y elegimos  $x_k$  como variable de entrada en la base, entonces se sigue que  $D_j^{p+1} \stackrel{L}{>} 0 \quad \forall j \in \Gamma^{p+1}$ .

*Demostración.* Si  $\Gamma_i^p = \emptyset$ , entonces  $x_i = \bar{b}_i^p + \sum_{j \in \Gamma^p} \bar{a}_{i,j}^p (-x_j) < 0$  para toda solución factible  $x$  de  $(PLE_R)$ , ya que  $\bar{b}_i^p < 0$  y,  $\bar{a}_{i,j}^p \geq 0 \quad \forall j \in \Gamma^p$  al suponer que  $\Gamma_i^p = \emptyset$ . Por tanto, se sigue que  $(PLE_R)$  es no factible.

Suponer ahora que  $\Gamma_i^p \neq \emptyset$ , entonces existe un  $k \in \Gamma_i^p$  que satisface (2.5) y, seleccionamos la variable  $x_k$  para entrar en la base. Aplicando las operaciones de pivotaje tenemos:

(i)  $D_i^{p+1} = -\frac{1}{\bar{a}_{i,k}^p} D_k^p \stackrel{L}{>} 0$ , ya que  $\bar{a}_{i,k}^p < 0$  al tomar  $k \in \Gamma_i^p$  y,  $D_k^p \stackrel{L}{>} 0$  por hipótesis al tener  $k \in \Gamma^p$ .

(ii)  $D_j^{p+1} = D_j^p - \frac{\bar{a}_{i,j}^p}{\bar{a}_{i,k}^p} D_k^p \quad \forall j \in \Gamma^p \setminus \{k\}$ . Dado un  $j \in \Gamma^p \setminus \{k\}$  tenemos dos casos:

- Si  $j \in \Gamma^p \setminus \Gamma_i^p$ , tenemos que  $D_j^{p+1} \stackrel{L}{>} 0$ , ya que  $D_j^p, D_k^p \stackrel{L}{>} 0$  por hipótesis al tomar  $j, k \in \Gamma^p$ . Por otra parte  $\bar{a}_{i,k}^p < 0$  y,  $\bar{a}_{i,j}^p \geq 0$  porque  $j \notin \Gamma_i^p$ .
- En caso contrario, si  $j \in \Gamma_i^p$ , por (2.5) tenemos

$$\frac{1}{\bar{a}_{i,k}^p} D_k^p \stackrel{L}{>} \frac{1}{\bar{a}_{i,j}^p} D_j^p.$$

Multiplicando ambos lados de la desigualdad por  $-\bar{a}_{i,j}^p > 0$ , se sigue que

$$-\frac{\bar{a}_{i,j}^p}{\bar{a}_{i,k}^p} D_k^p \stackrel{L}{>} -D_j^p.$$

Y por tanto, se obtiene que  $D_j^{p+1} = D_j^p - \frac{\bar{a}_{i,j}^p}{\bar{a}_{i,k}^p} D_k^p \stackrel{L}{>} 0$ .

Así pues, hemos probado que  $D_j^{p+1} \stackrel{L}{>} 0 \quad \forall j \in \Gamma^p \setminus \{k\}$ .

Por último, notar que  $\Gamma^{p+1} = (\Gamma^p \setminus \{k\}) \cup \{i\}$  debido a los cambios realizados en la base. Por tanto, por los puntos (i) y (ii), se sigue que  $D_j^{p+1} \stackrel{L}{>} 0 \quad \forall j \in \Gamma^{p+1}$ , probando la proposición.  $\square$

**Teorema 2.8.** *Si empezamos con una SBPFD  $(z, x)$  del problema  $(PLE_R)$  verificando  $\overline{D}_j^0 \stackrel{L}{>} 0 \quad \forall j \in \Gamma^0$  y aplicamos el algoritmo simplex dual lexicográfico, entonces en un número finito de paso probamos que  $(PLE_R)$  es no factible o encontramos la solución óptima del problema.*

*Demostración.* Si en alguna iteración del algoritmo simplex dual lexicográfico se prueba por la proposición 2.7 que  $(PLE_R)$  es no factible, entonces queda probado el teorema. En caso contrario, suponer que  $(PLE_R)$  es factible, entonces la sucesión  $\{D_0^p\}_{p \geq 0}$  obtenida en cada iteración es, por la proposición 2.6, lexicográficamente estrictamente decreciente y, de esta forma, nunca se repite base y evitamos la posibilidad de ciclos. Finalmente, como el algoritmo simplex dual recorre como máximo  $\binom{n}{m}$  puntos, entonces tenemos que obtener su solución óptima en un número finito de pasos.  $\square$

## 2.4. Método fraccional de Gomory para problemas lineales enteros puros

El método fraccional de Gomory, propuesto por dicho autor en Gomory [4], es un algoritmo diseñado para resolver problemas de programación lineal entera pura  $(PLE)$ . El algoritmo se basa en resolver una secuencia de problemas lineales relajados asociados a los que se les añaden sucesivos *cortes de Gomory* con objeto de ir acotando el recinto de factibilidad de  $(PLE_R)$ , de forma que la solución óptima de  $(PLE)$  pase a ser un punto extremo del recinto modificado y, el algoritmo simplex dual lexicográfico obtenga la solución óptima de  $(PLE)$ . En caso contrario, el algoritmo prueba que  $(PLE)$  es no factible.

A partir de este apartado en adelante, denotaremos por  $D_j^t$  a la columna asociada a la variable no básica  $x_j^t$  y, por  $\Gamma^t$  al conjunto de los índices de las variables no básicas de la tabla óptima correspondiente a la  $t$ -ésima iteración del algoritmo fraccional de Gomory.

El método fraccional de Gomory viene dado por el siguiente algoritmo.

---

### Algoritmo 1 Algoritmo fraccional de Gomory para problemas $(PLE)$

---

1: **Inicializamos:**

$t \leftarrow 1$

$S_R^1 \leftarrow \{x \in \mathbb{R}_+^n : Ax = b\}$

2: **while** 1 **do**

3:     **Paso 1:** Resolvemos el problema relajado aplicando el algoritmo simplex dual lexicográfico.

4:

$$\begin{aligned} \text{Máx} \quad & z^t(x) = c^T x \\ \text{s.a:} \quad & x \in S_R^t \end{aligned} \quad (PLE_R^t)$$

5:     **Paso 2:** Test de infactibilidad.

6:         **if**  $(PLE_R^t)$  es no factible **then**

7:              $(PLE)$  es también no factible.

8:             **Stop**

9:         **end if**

10:     Sea  $(z^t, x^t)$  la solución óptima de  $(PLE_R^t)$ .

11:     **Paso 3:** Test de optimalidad.

12:         **if**  $(z^t, x^t) \in \mathbb{Z}^{n+1}$  **then**

13:              $(z^t, x^t)$  es solución de  $(PLE)$ .

14:             **Stop**

15:         **end if**

---

---

16: **Paso 4:** Añadimos un corte de Gomory (en el punto 2.4.1 se muestra su construcción).  
17: Sea  $\Delta^t = \{x \in \mathbb{R}^n : \sum_{j \in \Gamma^t} (-f_{i,j}^t)(-x_j) = f_{i,0}^t\}$  un corte de Gomory correspondiente al problema  $(PLE_R^t)$ .  
18:  $S_R^{t+1} \leftarrow S_R^t \cap \Delta^t$   
19: **Paso 5:**  
20:  $t \leftarrow t + 1$   
21: **end while**

---

Denotamos por  $S^0$  la región de factibilidad del problema  $(PLE)$ , de modo que

$$S^0 = \{x \in \mathbb{Z}_+^n : Ax = b\},$$

entonces las regiones de factibilidad  $S_R^{t+1}$  obtenidas en las iteraciones verifican:

- (i)  $S_R^{t+1} \subseteq S_R^t \quad \forall t \geq 1$ .
- (ii)  $S^0 \subseteq S_R^t \quad \forall t \geq 1$ , ya que  $S_R^t = S_R^1 \cap (\cap_{i=1}^{t-1} \Delta^i)$  y aplicando la definición de corte.

Se mostrará más adelante que, en un número finito de pasos, el algoritmo encuentra el mayor elemento lexicográfico de  $(PLE)$  o prueba que  $(PLE)$  es no factible.

### 2.4.1. Corte fraccional de Gomory

Suponer que nos encontramos en la primera iteración del algoritmo fraccional de Gomory y, hemos resuelto el problema relajado  $(PLE_R^1)$  obteniendo la solución óptima  $(z^1, x^1)$ . Suponer que  $(z^1, x^1) \notin Z^{n+1}$  y que  $x_i^1$  es la variable básica cuyo índice alcanza el siguiente mínimo:

$$i = \min_{0 \leq j \leq m} \{j : b_j^1 \notin \mathbb{Z}\},$$

donde  $b_0^1 = \bar{z}^1$ . Haciendo uso de este cambio de notación y denotando  $x_0^1 = z^1$ ,  $\bar{a}_{0,j}^1 = \bar{c}_j^1 \quad \forall j \in \Gamma^1$ , suponer que la tabla óptima asociada a esta solución es

<i>Variables</i>	<i>Términos constantes</i>	$-x_{m+1}^1$	$\dots$	$-x_n^1$
$x_0^1$	$\bar{b}_0^1$	$\bar{a}_{0,m+1}^1$	$\dots$	$\bar{a}_{0,n}^1$
$x_1^1$	$\bar{b}_1^1$	$\bar{a}_{1,m+1}^1$	$\dots$	$\bar{a}_{1,n}^1$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$x_i^1$	$\bar{b}_i^1 \notin \mathbb{Z}$	$\bar{a}_{i,m+1}^1$	$\dots$	$\bar{a}_{i,n}^1$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$x_m^1$	$\bar{b}_m^1$	$\bar{a}_{m,m+1}^1$	$\dots$	$\bar{a}_{m,n}^1$
$x_{m+1}^1$	0	-1	$\dots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$x_n^1$	0	0	$\dots$	-1

Tabla 2.3: Tabla óptima del problema  $(PLE_R^1)$ .

Por la Tabla 2.3, la fila asociada a  $x_i^1$  se corresponde con la restricción

$$x_i^1 = \bar{b}_i^1 + \sum_{j \in \Gamma^1} \bar{a}_{i,j}^1 (-x_j^1). \quad (2.6)$$



Como el valor de  $x_i^1$  en la solución es  $\bar{b}_i^1$  y  $\bar{b}_i^1 \notin \mathbb{Z}$ , se sigue necesariamente que  $\bar{b}_i^1 > 0$  ya que  $x_i^1 \in \mathbb{R}^+$ . Sea  $\lfloor \cdot \rfloor : \mathbb{R} \rightarrow \mathbb{Z}$  la función *suelo entero* dada por

$$\lfloor \gamma \rfloor = \max\{k \in \mathbb{Z} : k \leq \gamma\}.$$

En general, se tiene  $0 \leq \gamma - \lfloor \gamma \rfloor < 1$  donde  $\gamma - \lfloor \gamma \rfloor$  recibe el nombre de *parte fraccional* de  $\gamma$ .

Definimos

$$f_{i,0}^1 = \bar{b}_i^1 - \lfloor \bar{b}_i^1 \rfloor \quad \text{y} \quad f_{i,j}^1 = \bar{a}_{i,j}^1 - \lfloor \bar{a}_{i,j}^1 \rfloor, \quad \forall j \in \Gamma^1.$$

Como  $\bar{b}_i^1 \notin \mathbb{Z}$ , se tiene que  $f_{i,0}^1 > 0$ . Sustituyendo  $\bar{b}_i^1$  y  $\bar{a}_{i,j}^1$  en (2.6), tenemos

$$\begin{aligned} x_i^1 &= f_{i,0}^1 + \lfloor \bar{b}_i^1 \rfloor + \sum_{j \in \Gamma^1} (f_{i,j}^1 + \lfloor \bar{a}_{i,j}^1 \rfloor)(-x_j^1) \\ &= f_{i,0}^1 + \lfloor \bar{b}_i^1 \rfloor + \sum_{j \in \Gamma^1} f_{i,j}^1(-x_j^1) + \sum_{j \in \Gamma^1} \lfloor \bar{a}_{i,j}^1 \rfloor(-x_j^1). \end{aligned}$$

Reorganizando términos, obtenemos la ecuación

$$-f_{i,0}^1 + \sum_{j \in \Gamma^1} f_{i,j}^1 x_j^1 = -x_i^1 + \lfloor \bar{b}_i^1 \rfloor + \sum_{j \in \Gamma^1} \lfloor \bar{a}_{i,j}^1 \rfloor(-x_j^1). \quad (2.7)$$

La expresión de la derecha de (2.7) es un número entero para toda solución factible de (PLE). Por tanto, dada una solución factible  $x$  de (PLE), se tiene que

$$-f_{i,0}^1 + \sum_{j \in \Gamma^1} f_{i,j}^1 x_j \in \mathbb{Z}. \quad (2.8)$$

Por otro lado, como  $x \geq 0$  y  $f_{i,j}^1 \geq 0 \quad \forall j \in \Gamma^1$ , tenemos que

$$-f_{i,0}^1 + \sum_{j \in \Gamma^1} f_{i,j}^1 x_j \geq -f_{i,0}^1. \quad (2.9)$$

Pero  $f_{i,0}^1 \in (0, 1)$ , luego por (2.8) y (2.9), se verifica que

$$-f_{i,0}^1 + \sum_{j \in \Gamma^1} f_{i,j}^1 x_j \in \mathbb{Z}^+. \quad (2.10)$$

Por tanto, acabamos de probar que para toda solución entera factible  $x$  de (PLE), se tiene que

$$-f_{i,0}^1 + \sum_{j \in \Gamma^1} (-f_{i,j}^1)(-x_j) \geq 0. \quad (2.11)$$

**Proposición 2.9.** *La restricción (2.11) es un corte y recibe el nombre de corte fraccional de Gomory asociado a la variable  $x_i^1$ .*

*Demostración.* El hiperplano

$$H = \{x \in \mathbb{R}^n : -f_{i,0}^1 + \sum_{j \in \Gamma^1} (-f_{i,j}^1)(-x_j) = 0\}$$

verifica:

- (i)  $x^1 \notin H^+$  para la solución óptima  $x^1$  de  $(PLE_R^1)$ , ya que  $-f_{i,0}^1 + \sum_{j \in \Gamma^1} (-f_{i,j}^1)(-x_j^1) = -f_{i,0}^1 < 0$ .
- (ii)  $x \in H^+$  para toda solución factible  $x$  de (PLE), por (2.11).

Por tanto,  $H$  es un plano de corte y (2.11) es su corte asociado. □

Una vez que hemos obtenido el corte de Gomory asociado a la variable  $x_i^1$ , por el cuatro paso del Algoritmo 1, añadimos al problema  $(PLE_R^1)$  la variable auxiliar  $s^1 \geq 0$  asociada al corte (2.11) dada por

$$s^1 = -f_{i,0}^1 + \sum_{j \in \Gamma^1} (-f_{i,j}^1)(-x_j^1), \quad (2.12)$$

donde, por (2.10),  $s^1 \in \mathbb{Z}^+$  para toda solución factible de  $(PLE)$ . De este modo, la región de factibilidad  $S_R^2$  del problema aumentado  $(PLE_R^2)$  no contiene la solución óptima  $(x_0^1, x^1)$  de  $(PLE_R^1)$ , pero sigue conteniendo todas las soluciones enteras factibles de  $(PLE)$ .

Añadimos la nueva restricción (2.12) al final de la Tabla 2.3 obteniendo la siguiente tabla.

<i>Variables</i>	<i>Términos constantes</i>	$-x_{m+1}^1$	$\dots$	$-x_n^1$
$x_0^1$	$\bar{b}_0^1$	$\bar{a}_{0,m+1}^1$	$\dots$	$\bar{a}_{0,n}^1$
$x_1^1$	$\bar{b}_1^1$	$\bar{a}_{1,m+1}^1$	$\dots$	$\bar{a}_{1,n}^1$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$x_i^1$	$\bar{b}_i^1 \notin \mathbb{Z}$	$\bar{a}_{i,m+1}^1$	$\dots$	$\bar{a}_{i,n}^1$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$x_m^1$	$\bar{b}_m^1$	$\bar{a}_{m,m+1}^1$	$\dots$	$\bar{a}_{m,n}^1$
$x_{m+1}^1$	0	-1	$\dots$	0
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$x_n^1$	0	0	$\dots$	-1
$s^1$	$-f_{i,0}^1$	$-f_{i,m+1}^1$	$\dots$	$-f_{i,n}^1$

Tabla 2.4: Tabla 2.3 aumentada resultante al añadir el corte  $s^1$ .

Por el funcionamiento del algoritmo simplex dual lexicográfico, la Tabla 2.4 preserva la factibilidad dual ya que la solución óptima  $(x_0^1, x^1)$  verificaba que  $D_j^1 \stackrel{L}{>} 0 \quad \forall j \in \Gamma^1$  y, por tanto, se sigue necesariamente que  $\bar{a}_{0,j}^1 \geq 0 \quad \forall j \in \Gamma^1$ . Sin embargo, la Tabla 2.4 no es factible primal ya que  $s^1 = -f_{i,0}^1 < 0$ . De esta manera, como  $(x_0^1, x^1, s^1)$  es una SBPFD para  $(PLE_R^2)$ , podemos aplicar el algoritmo simplex dual lexicográfico. Para la primera iteración, elegimos  $s^1$  como variable de salida y como variable de entrada aquella dada por la proposición 2.7. De esta forma, después de realizar este primer cambio de base,  $s^1$  se convierte en una variable no básica tomando el valor cero en la siguiente solución.

Una vez resuelto el problema  $(PLE_R^2)$ , seguimos los pasos del método fraccional de Gomory descritos en el Algoritmo 1 hasta obtener la solución óptima de  $(PLE)$  o probar que  $(PLE)$  es no factible. La convergencia del algoritmo se mostrará en el siguiente punto.

### 2.4.2. Convergencia finita del método fraccional de Gomory

Vamos a probar la convergencia finita del algoritmo del método fraccional de Gomory para problemas lineales enteros puros  $(PLE)$ .

Suponer que nos encontramos en la  $t$ -ésima iteración del algoritmo fraccional de Gomory y hemos obtenido la solución óptima  $(z^t, x^t)$  del problema  $(PLE_R^t)$ , de manera que  $(z^t, x^t) \notin \mathbb{Z}^{n+1}$ . Por convenio en los siguientes resultados, hacemos uso de la notación ya empleada en el punto 2.4.1 de manera que  $(x_0^t, x^t) = (z^t, x^t)$ ,  $\bar{a}_{0,j}^t = \bar{c}_j^t \quad \forall j \in \Gamma^t$  y, denotamos además  $\bar{a}_{i,0}^t = \bar{b}_i^t \quad \forall i = 0, \dots, m$ .

**Proposición 2.10.** *Sea  $(x_0^t, x^t)$  la solución óptima de  $(PLE_R^t)$ , suponer que  $x_i^t \in \mathbb{Z} \quad \forall i = 0, \dots, s-1$  y  $x_s^t \notin \mathbb{Z}$  con  $s \leq n$ . Definimos el vector  $\alpha^t = (x_0^t, \dots, x_{s-1}^t, \lfloor x_s^t \rfloor, d, \dots, d) \in \mathbb{Z}^{n+1}$ , donde  $d \in \mathbb{Z}^+$  viene dado por (2.1). Si  $(x_0, x)$  es una solución factible cualquiera del problema  $(PLE)$ , entonces  $(x_0, x) \stackrel{L}{\leq} \alpha^t$ .*

*Demostración.* Dado  $(x_0, x)$  una solución factible de  $(PLE)$ , tenemos que  $x \in S^0 \subseteq S_R^1$ , por construcción de las regiones de factibilidad en el Algoritmo 1. Como  $(x_0^t, x^t)$  es solución óptima de  $(PLE_R^t)$ , por la proposición (2.5), se sigue que  $(x_0^t, x^t)$  es la mayor solución factible lexicográfica de  $(PLE_R^t)$ . Y por tanto, se tiene que  $(x_0, x) \stackrel{L}{\leq} (x_0^t, x^t)$ . Por definición de orden lexicográfico  $(\stackrel{L}{\leq})$ , tenemos dos casos:

- (i) Si  $(x_0, x) = (x_0^t, x^t)$ , tenemos una contradicción, ya que  $x_s^t \notin \mathbb{Z}$ .
- (ii) Necesariamente, se debe de cumplir que  $(x_0, x) \stackrel{L}{<} (x_0^t, x^t)$ . Por tanto,  $\exists k \ni x_k < x_k^t \wedge x_i = x_i^t \forall i < k$ . Tomando el vector  $\alpha^t$  dado, distinguimos los posibles valores que puede tomar el índice  $k$ :
  - Si  $k = 0, \dots, s-1$ , es inmediato por definición que  $(x_0, x) \stackrel{L}{<} \alpha^t$ .
  - Si  $k = s$ , entonces  $x_s < x_s^t$ . Como  $x_s^t \notin \mathbb{Z}$ , se sigue que  $x_s \leq \lfloor x_s^t \rfloor < x_s^t$ . De esta forma, si  $x_s < \lfloor x_s^t \rfloor$ , por definición, tenemos  $(x_0, x) \stackrel{L}{<} \alpha^t$ . En caso contrario, si  $x_s = \lfloor x_s^t \rfloor$ , como  $x \in S^0 \subseteq S_R^1$ , por (2.1) tenemos que  $x_j \leq d \forall j = s+1, \dots, n$ . Y por tanto,  $(x_0, x) \stackrel{L}{\leq} \alpha^t$ .
  - Si  $k \geq s+1$ , análogamente, por (2.1) tenemos que  $x_j \leq d \forall j = s+1, \dots, n$  y se verifica que  $(x_0, x) \stackrel{L}{\leq} \alpha^t$ .

Por tanto, hemos probado que  $(x_0, x) \stackrel{L}{\leq} \alpha^t$ . □

**Proposición 2.11.** Sea  $(x_0^t, x^t)$  la solución óptima de  $(PLE_R^t)$  y sea  $\alpha^t$  el vector definido en la proposición 2.10. Si añadimos el corte de Gomory  $s^t = -f_{s,0}^t + \sum_{j \in \Gamma^t} (-f_{s,j}^t)(-x_j^t)$  asociado a la variable  $x_s^t$  y resolvemos el problema aumentado  $(PLE_R^{t+1})$ , entonces la solución óptima obtenida verifica que  $(x_0^{t+1}, x^{t+1}) \stackrel{L}{<} \alpha^t$ .

*Demostración.* Es suficiente probar que la SBPFD  $(\hat{x}_0^t, \hat{x}^t)$  de  $(PLE_R^{t+1})$  obtenida en la primera iteración del algoritmo simplex dual lexicográfico verifica que  $(\hat{x}_0^t, \hat{x}^t) \stackrel{L}{<} \alpha^t$ . De este modo, si  $(\hat{x}_0^t, \hat{x}^t)$  es solución óptima de  $(PLE_R^{t+1})$ , habremos probado la proposición. En caso contrario, por la proposición 2.6, tendremos que  $(x_0^{t+1}, x^{t+1}) \stackrel{L}{<} (\hat{x}_0^t, \hat{x}^t) \stackrel{L}{<} \alpha^t$ , probando también la proposición.

La tabla aumentada resultante al añadir el corte de Gomory tiene la forma a la mostrada en la Tabla 2.4 y, es factible dual verificando  $D_j^t > 0 \forall j \in \Gamma^t$ . Realizamos la primera iteración del algoritmo simplex dual lexicográfico eligiendo  $s^t$  como variable de salida, ya que  $s^t = -f_{s,0}^t < 0$  y  $x^t \geq 0$ . Para un  $k \in \Gamma^t$ , sea  $x_k$  la variable que entra en la base. Suponer que  $(\hat{x}_0^t, \hat{x}^t)$  es la SBPFD obtenida después de realizar la primera iteración, entonces aplicando las operaciones de pivoteaje tenemos que

$$\begin{pmatrix} \hat{x}_0^t \\ \hat{x}^t \end{pmatrix} = \begin{pmatrix} x_0^t \\ x^t \end{pmatrix} - \frac{f_{s,0}^t}{f_{s,k}^t} D_k^t, \quad (2.13)$$

donde  $D_k^t > 0$ ,  $f_{s,0}^t > 0$  por construcción del corte y,  $-f_{s,k}^t < 0$  por asumir que la variable  $x_k$  entra en la base siguiendo el criterio de elección de la proposición 2.7. A continuación, vamos a probar que  $(\hat{x}_0^t, \hat{x}^t) \stackrel{L}{<} \alpha^t = (x_0^t, \dots, x_{s-1}^t, \lfloor x_s^t \rfloor, d, \dots, d)$ . Como  $D_k^t > 0$ , por definición,  $\exists q \ni \bar{a}_{q,k}^t > 0 \wedge \bar{a}_{i,k}^t = 0 \forall i < q$ . Así pues, distinguimos los posibles valores que puede tomar el índice  $q$ :

- (i) Si  $q = 0, \dots, s-1$ , entonces se sigue por (2.13) que

$$\hat{x}_i^t = x_i^t \forall i = 0, \dots, q-1.$$

Por otra parte, como  $f_{s,0}^t > 0$ ,  $f_{s,k}^t > 0$  y  $\bar{a}_{q,k}^t > 0$ , se tiene además

$$\hat{x}_q^t = x_q^t - \frac{f_{s,0}^t}{f_{s,k}^t} \bar{a}_{q,k}^t < x_q^t.$$

Por tanto, acabamos de probar que  $(\hat{x}_0^t, \hat{x}^t) \stackrel{L}{<} \alpha^t$ .

- (ii) Si  $q \geq s$ , entonces  $\bar{a}_{i,k}^t = 0 \ \forall i = 0, \dots, q-1$  y, como  $q-1 \geq s-1$ , podemos asegurar que  $\bar{a}_{i,k}^t = 0 \ \forall i = 0, \dots, s-1$ . Por tanto, por (2.13) se sigue que

$$\hat{x}_i^t = x_i^t \ \forall i = 0, \dots, s-1. \quad (2.14)$$

Por otro lado, notar que

$$\bar{a}_{s,k}^t \geq \bar{a}_{s,k}^t - \lfloor \bar{a}_{s,k}^t \rfloor = f_{s,k}^t.$$

Dividimos ambos lados de la desigualdad por  $f_{s,k}^t > 0$ , obteniendo

$$\frac{\bar{a}_{s,k}^t}{f_{s,k}^t} \geq 1.$$

Multiplicando ambos lados por  $-f_{s,0}^t < 0$ , queda

$$-\frac{f_{s,0}^t}{f_{s,k}^t} \bar{a}_{s,k}^t \leq -f_{s,0}^t.$$

Por tanto, aplicando esta desigualdad a (2.13), se tiene para la variable  $\hat{x}_s^t$  que

$$\hat{x}_s^t = x_s^t - \frac{f_{s,0}^t}{f_{s,k}^t} \bar{a}_{s,k}^t \leq x_s^t - f_{s,0}^t,$$

y como  $f_{s,0}^t = \bar{a}_{s,0}^t - \lfloor \bar{a}_{s,0}^t \rfloor$ , entonces obtenemos que

$$\hat{x}_s^t \leq x_s^t - \bar{a}_{s,0}^t + \lfloor \bar{a}_{s,0}^t \rfloor.$$

Por último, teniendo en cuenta que  $\bar{a}_{s,0}^t$  es el valor de  $x_s^t$  en la solución, se sigue

$$\hat{x}_s^t \leq x_s^t - x_s^t + \lfloor x_s^t \rfloor = \lfloor x_s^t \rfloor.$$

Si se tiene que  $\hat{x}_s^t < \lfloor x_s^t \rfloor$ , junto con (2.14), habremos probado que  $(\hat{x}_0^t, \hat{x}^t) \stackrel{L}{<} \alpha^t$ . En caso contrario, si  $\hat{x}_s^t = \lfloor x_s^t \rfloor$ , necesariamente se debe de cumplir que  $\bar{a}_{s,k}^t = 0$ .

En efecto, como  $q \geq s$ , entonces se debe de verificar que  $\bar{a}_{s,k}^t = 0$  o  $\bar{a}_{s,k}^t > 0$ . Suponer que  $\bar{a}_{s,k}^t > 0$ , al estar asumiendo que  $\hat{x}_s^t = \lfloor x_s^t \rfloor$ , por (2.13) se sigue

$$\lfloor x_s^t \rfloor = \hat{x}_s^t = x_s^t - \frac{f_{s,0}^t}{f_{s,k}^t} \bar{a}_{s,k}^t.$$

Pero  $\bar{a}_{s,k}^t > 0$ , luego

$$\lfloor x_s^t \rfloor = x_s^t - \frac{f_{s,0}^t}{f_{s,k}^t} \bar{a}_{s,k}^t < x_s^t,$$

obteniendo una contradicción. Así pues, se debe de cumplir que  $\bar{a}_{s,k}^t = 0$ . De esta forma, tenemos que  $q \geq s+1$ . Notar además que como  $x^t \in S_R^t \subseteq S_R^1$ , por (2.1), se sigue que  $x_i^t \leq d \ \forall i \geq s+1$ .

Así pues, por (2.13) tenemos:

- $\hat{x}_i^t = x_i^t \leq d \ \forall i = s+1, \dots, q-1$ .
- $\hat{x}_q^t = x_q^t - \frac{f_{s,0}^t}{f_{s,k}^t} \bar{a}_{q,k}^t < x_q^t \leq d$ .

Y por tanto, junto con (2.14), se sigue que  $(\hat{x}_0^t, \hat{x}^t) \stackrel{L}{<} \alpha^t$ .

De esta manera, hemos probado que para cualquier valor de  $q \geq 0$  se tiene que  $(\hat{x}_0^q, \hat{x}^q) \stackrel{L}{<} \alpha^t$ . Y por tanto, por lo comentado al principio de la demostración, hemos probado que  $(x_0^{t+1}, x^{t+1}) \stackrel{L}{<} \alpha^t$ .  $\square$

**Corolario 2.12.** Si la solución óptima  $(x_0^{t+1}, x^{t+1}) \notin \mathbb{Z}^{n+1}$ , entonces  $\alpha^{t+1} \stackrel{L}{<} \alpha^t$ .

*Demostración.* Como  $(x_0^{t+1}, x^{t+1}) \notin \mathbb{Z}^{n+1}$ , entonces  $\exists s \in \{0, 1, \dots, n\} \ni x_s^{t+1} \in \mathbb{Z} \ \forall i < s$  y  $x_s^{t+1} \notin \mathbb{Z}$ . De esta forma, se sigue que  $\lfloor x_s^{t+1} \rfloor < x_s^{t+1}$  y, por tanto

$$\alpha^{t+1} = (x_0^{t+1}, \dots, x_{s-1}^{t+1}, \lfloor x_s^{t+1} \rfloor, d, \dots, d) \stackrel{L}{<} (x_0^{t+1}, x^{t+1}).$$

Además, por la proposición 2.11 se tiene que  $(x_0^{t+1}, x^{t+1}) \stackrel{L}{<} \alpha^t$ , luego hemos probamos finalmente que  $\alpha^{t+1} \stackrel{L}{<} \alpha^t$ .  $\square$

A continuación, presentamos el resultado más importante de este capítulo que es la convergencia finita del método fraccional de Gomory.

**Teorema 2.13. Convergencia finita del método fraccional de Gomory.** Aplicamos el algoritmo fraccional de Gomory eligiendo cada corte asociado a la variable no entera de menor índice y, resolvemos el problema aumentado aplicando el algoritmo simplex dual lexicográfico, entonces en un número finito de iteraciones, el algoritmo encuentra una solución óptima del problema lineal entero puro (PLE) o prueba que es no factible.

*Demostración.* Sean  $c_j^+ = \max\{0, c_j\}$ ,  $c_j^- = \min\{0, c_j\}$ . Definimos los vectores  $\underline{\alpha}, \overline{\alpha} \in \mathbb{Z}^{n+1}$  dados por

$$\begin{aligned} \underline{\alpha} &= (d \sum_{j=1}^n c_j^-, 0, \dots, 0), \\ \overline{\alpha} &= (d \sum_{j=1}^n c_j^+, d, \dots, d). \end{aligned}$$

Para demostrar el teorema, necesitamos primero los siguientes resultados:

- (i) Para toda solución factible  $(x_0, x)$  de (PLE), tenemos que

$$d \sum_{j=1}^n c_j^- \leq \sum_{j=1}^n c_j^- x_j \leq \sum_{j=1}^n c_j x_j = x_0 \leq \sum_{j=1}^n c_j^+ x_j \leq d \sum_{j=1}^n c_j^+.$$

Por tanto, se sigue que

$$\underline{\alpha} \stackrel{L}{\leq} (x_0, x) \stackrel{L}{\leq} \overline{\alpha}. \quad (2.15)$$

- (ii) Suponer que la solución óptima  $(x_0^1, x^1)$  de  $(PLE_R^1)$  no verifica las condiciones de integridad, entonces se tiene que

$$\alpha^1 \stackrel{L}{<} \overline{\alpha}. \quad (2.16)$$

En efecto, por definición de  $\alpha^1$  en la proposición 2.10 se tiene que  $\alpha^1 \stackrel{L}{<} (x_0^1, x^1)$  y,  $(x_0^1, x^1) \stackrel{L}{\leq} \overline{\alpha}$  por un razonamiento similar al de (i), con lo que se sigue la desigualdad.

- (iii) Para todo  $t \geq 1$  se sigue que

$$\alpha^t \stackrel{L}{\leq} \alpha^t, \quad (2.17)$$

siguiendo un razonamiento similar al de (i).

(iv) El número de vectores  $y \in \mathbb{Z}^{n+1}$  tales que

$$\begin{aligned} d \sum_{j=1}^n c_j^- \leq y_0 \leq d \sum_{j=1}^n c_j^+, \\ 0 \leq y_j \leq d \quad \forall j = 1, \dots, n. \end{aligned} \quad (2.18)$$

que verifican las desigualdades

$$\underline{\alpha} \stackrel{L}{\leq} y \stackrel{L}{\leq} \bar{\alpha}, \quad (2.19)$$

es de  $(d+1)^n \left[ d \sum_{j=1}^n (c_j^+ - c_j^-) + 1 \right]$ .

Una vez que tenemos todos estos resultados podemos probar el teorema.

Sea  $M = (d+1)^n \left[ d \sum_{j=1}^n (c_j^+ - c_j^-) + 1 \right]$  y  $M^- = M - 1$ . Si en alguna iteración  $t \leq M$  del algoritmo obtenemos que el problema  $(PLE_R^t)$  es no factible, entonces  $(PLE)$  es también no factible probando el teorema. Ahora suponer que  $(PLE)$  es factible, si en alguna iteración  $t \leq M^-$  el algoritmo de Gomory encuentra la solución óptima entera, entonces hemos probado el teorema. En caso contrario, suponer que nos encontramos en la  $M^-$ -ésima iteración y que la solución de  $(PLE_R^{M^-})$   $(x_0^{M^-}, x^{M^-}) \notin \mathbb{Z}^{n+1}$ . Vamos a probar que en este caso el algoritmo encuentra una solución óptima en la  $M$ -ésima iteración.

Por el corolario 2.12, la sucesión que obtenemos en cada iteración del algoritmo  $\{\alpha^t\}_{t=1}^{M^-}$  es lexicográficamente estrictamente decreciente. Además, por (2.16) tenemos que  $\alpha^1 \stackrel{L}{<} \bar{\alpha}$  y, como  $\alpha^t \in \mathbb{Z}^{n+1}$  verifica las condiciones (2.18) y (2.19) para todo  $t \geq 1$ , entonces se debe de seguir necesariamente que  $\underline{\alpha} = \alpha^{M^-}$ , ya que acabamos de ver que hay  $M$  vectores en  $\mathbb{Z}^{n+1}$  verificando estas dos condiciones. Por otra parte, por la proposición 2.10, tenemos que toda solución factible  $(x_0, x)$  de  $(PLE)$  verifica que  $(x_0, x) \stackrel{L}{\leq} \alpha^{M^-}$ . De esta manera, se sigue necesariamente que  $(x_0, x) = \alpha^{M^-}$ , ya que en caso contrario se sigue que  $(x_0, x) \stackrel{L}{<} \alpha^{M^-} = \underline{\alpha}$  contradiciendo la desigualdad (2.15). Además, como estamos asumiendo que  $(x_0^{M^-}, x^{M^-})$  es no entero, por la proposición 2.11, tenemos que  $(x_0^M, x^M) \stackrel{L}{<} \alpha^{M^-}$ . Distinguimos dos casos: (1) si  $(x_0^M, x^M) \in \mathbb{Z}^{n+1}$ , ya hemos probado el teorema. (2) En caso contrario, como  $(x_0^M, x^M) \notin \mathbb{Z}^{n+1}$  tenemos que  $\alpha^M \stackrel{L}{<} (x_0^M, x^M)$  y además  $\alpha^{M^-} = (x_0, x)$ . Por tanto, se sigue que  $\alpha^M \stackrel{L}{<} (x_0, x)$ , obteniendo una contradicción por la proposición 2.10. Así pues, se sigue necesariamente que  $(x_0^M, x^M) \in \mathbb{Z}^{n+1}$  y, es la solución óptima de  $(PLE)$  probando el teorema. Es más, acabamos de probar que el algoritmo fraccional de Gomory necesita como máximo  $M$  iteraciones para encontrar una solución óptima del problema lineal entero  $(PLE)$  o probar que es no factible.  $\square$

## 2.5. Corte fraccional de Gomory para problemas lineales enteros mixtos

El método fraccional de Gomory se extiende directamente para la resolución de problemas lineales enteros mixtos. Dedicamos este punto para mostrar cómo se deriva la formulación de los cortes fraccionales de Gomory en este tipo de problemas.

Considerar el problema lineal entero mixto

$$\begin{aligned} \text{Máx} \quad & z(x) = c^T x \\ \text{s.a:} \quad & Ax = b \\ & x \in \mathbb{R}_+^n \\ & x_j \in \mathbb{Z}_+^n \quad \forall j \in J \end{aligned} \quad (PLEM)$$

donde  $J \subset \{1, 2, \dots, n\}$  es el conjunto de los índices correspondientes a las variables restringidas a tomar valores enteros.

Resolvemos su problema relajado asociado mediante el algoritmo simplex dual lexicográfico, obteniendo la solución óptima  $(z, x)$ . Suponer que la tabla óptima correspondiente es la Tabla 2.1. Sea  $\Gamma = \{m+1, \dots, n\}$  el conjunto de los índices correspondientes a las variables no básicas y, suponer que existe una variable básica  $x_i$ , con  $i \in J$ , tal que  $b_i \notin \mathbb{Z}$ . Por la Tabla 2.1, la fila asociada a  $x_i$  se corresponde con la restricción

$$x_i = \bar{b}_i + \sum_{j \in \Gamma} \bar{a}_{i,j}(-x_j). \quad (2.20)$$

De la misma manera a la realizada en el punto 2.4.1, definimos

$$f_{i,0} = \bar{b}_i - \lfloor \bar{b}_i \rfloor \quad \text{y} \quad f_{i,j} = \bar{a}_{i,j} - \lfloor \bar{a}_{i,j} \rfloor, \quad \forall j \in \Gamma \cap J,$$

de manera que  $0 < f_{i,0} < 1$  y  $0 \leq f_{i,j} < 1 \quad \forall j \in \Gamma \cap J$ .

Definimos los conjuntos

$$H = \Gamma \cap J, \quad F = \Gamma \setminus J,$$

y

$$P = \{j \in F : \bar{a}_{i,j} \geq 0\},$$

$$Q = \{j \in F : \bar{a}_{i,j} < 0\}.$$

Como  $\Gamma$  es un conjunto finito y la familia  $\{F, H\}$  forma una partición de  $\Gamma$  (esto es, se verifican  $\Gamma = F \cup H$ ,  $F \cap H = \emptyset$ ), podemos reescribir (2.20), sustituyendo  $\bar{b}_i$ , como

$$x_i = f_{i,0} + \lfloor \bar{b}_i \rfloor + \sum_{j \in F} \bar{a}_{i,j}(-x_j) + \sum_{j \in H} \bar{a}_{i,j}(-x_j). \quad (2.21)$$

Además, sustituyendo  $\bar{a}_{i,j} \quad \forall j \in H$  en (2.21), tenemos

$$\begin{aligned} x_i &= f_{i,0} + \lfloor \bar{b}_i \rfloor + \sum_{j \in F} \bar{a}_{i,j}(-x_j) + \sum_{j \in H} (f_{i,j} + \lfloor \bar{a}_{i,j} \rfloor)(-x_j) \\ &= f_{i,0} + \lfloor \bar{b}_i \rfloor + \sum_{j \in F} \bar{a}_{i,j}(-x_j) + \sum_{j \in H} f_{i,j}(-x_j) + \sum_{j \in H} \lfloor \bar{a}_{i,j} \rfloor(-x_j). \end{aligned}$$

Finalmente, reorganizando términos, la ecuación (2.20) se puede expresar como

$$\sum_{j \in H} f_{i,j}x_j + \sum_{j \in F} \bar{a}_{i,j}x_j = u + f_{i,0}, \quad (2.22)$$

donde

$$u = -x_i + \lfloor \bar{b}_i \rfloor + \sum_{j \in H} \lfloor \bar{a}_{i,j} \rfloor(-x_j).$$

Por otra parte, se tiene que  $u \in \mathbb{Z}$  para toda solución factible de (PLEM). Por tanto, se debe de verificar que  $u \geq 0$  o  $u \leq -1$ . Distinguiamos ambos casos:

(i) Si  $u \geq 0$ , por (2.22) obtenemos

$$\sum_{j \in H} f_{i,j}x_j + \sum_{j \in F} \bar{a}_{i,j}x_j \geq f_{i,0}. \quad (2.23)$$

Como  $F$  es un conjunto finito y la familia  $\{Q, P\}$  forma una partición de  $F$ , se tiene

$$\sum_{j \in F} \bar{a}_{i,j}x_j = \sum_{j \in Q} \bar{a}_{i,j}x_j + \sum_{j \in P} \bar{a}_{i,j}x_j,$$

y como  $\sum_{j \in Q} \bar{a}_{i,j}x_j \leq 0$ , se sigue por (2.23)

$$\sum_{j \in H} f_{i,j}x_j + \sum_{j \in P} \bar{a}_{i,j}x_j \geq \sum_{j \in H} f_{i,j}x_j + \sum_{j \in P} \bar{a}_{i,j}x_j + \sum_{j \in Q} \bar{a}_{i,j}x_j \geq f_{i,0},$$

obteniendo la desigualdad

$$\sum_{j \in H} f_{i,j}x_j + \sum_{j \in P} \bar{a}_{i,j}x_j \geq f_{i,0}. \quad (2.24)$$

(ii) Si  $u \leq -1$ , por (2.22) tenemos

$$\sum_{j \in H} f_{i,j}x_j + \sum_{j \in F} \bar{a}_{i,j}x_j \leq -1 + f_{i,0}.$$

Además, como  $f_{i,j} \geq 0$ ,  $x_j \geq 0$ ,  $\forall j \in H$ , se sigue que

$$\sum_{j \in F} \bar{a}_{i,j}x_j \leq \sum_{j \in H} f_{i,j}x_j + \sum_{j \in F} \bar{a}_{i,j}x_j \leq -1 + f_{i,0}.$$

Al igual que antes, como  $F$  es un conjunto finito y la familia  $\{Q, P\}$  forma una partición de  $F$ , se tiene

$$\sum_{j \in Q} \bar{a}_{i,j}x_j + \sum_{j \in P} \bar{a}_{i,j}x_j = \sum_{j \in F} \bar{a}_{i,j}x_j \leq -1 + f_{i,0}.$$

Ahora, multiplicamos ambos lados por  $f_{i,0} > 0$ , obteniendo

$$\sum_{j \in Q} \bar{a}_{i,j}x_j f_{i,0} + \sum_{j \in P} \bar{a}_{i,j}x_j f_{i,0} \leq f_{i,0}(-1 + f_{i,0}),$$

y como  $\bar{a}_{i,j} \geq 0$ ,  $x_j \geq 0$ ,  $\forall j \in P$ ,  $f_{i,0} > 0$ , se tiene que

$$\sum_{j \in Q} \bar{a}_{i,j}x_j f_{i,0} \leq \sum_{j \in Q} \bar{a}_{i,j}x_j f_{i,0} + \sum_{j \in P} \bar{a}_{i,j}x_j f_{i,0} \leq f_{i,0}(-1 + f_{i,0}).$$

Finalmente, multiplicando ambos lados por  $\frac{-1}{1-f_{i,0}} \in (-1, 0)$  obtenemos que

$$\sum_{j \in Q} \frac{(-\bar{a}_{i,j})x_j f_{i,0}}{(1-f_{i,0})} \geq f_{i,0}. \quad (2.25)$$

De esta manera, si  $u \geq 0$  se verifica la desigualdad (2.24) y, si  $u \leq -1$  se verifica la desigualdad (2.25). Es decir, necesariamente una de las dos desigualdades se debe de verificar. Además, las expresiones de la izquierda de (2.24), (2.25) son no negativas para toda solución factible de (PLEM).

Por tanto, tenemos que

$$-f_{i,0} + \sum_{j \in H} (-f_{i,j})(-x_j) + \sum_{j \in P} (-\bar{a}_{i,j})(-x_j) + \sum_{j \in Q} \frac{(-x_j)\bar{a}_{i,j}f_{i,0}}{(1-f_{i,0})} \geq 0, \quad (2.26)$$

para toda solución factible del problema (PLEM).

**Proposición 2.14.** La restricción (2.26) es un corte y recibe el nombre de corte fraccional de Gomory para problemas lineales enteros mixtos asociado a la variable  $x_i$ .

*Demostración.* Sea la función  $r: \mathbb{R}^n \rightarrow \mathbb{R}$  dada por

$$r(x) = -f_{i,0} + \sum_{j \in H} (-f_{i,j})(-x_j) + \sum_{j \in P} (-\bar{a}_{i,j})(-x_j) + \sum_{j \in Q} \frac{(-x_j)\bar{a}_{i,j}f_{i,0}}{(1-f_{i,0})},$$

entonces el hiperplano

$$H = \{x \in \mathbb{R}^n : r(x) = 0\}$$

verifica:

- (i)  $x \notin H^+$  para la solución óptima  $x$  de (PLEM), ya que  $r(x) = -f_{i,0} < 0$ .
- (ii)  $\tilde{x} \in H^+$  para toda solución factible  $\tilde{x}$  de (PLEM), por (2.26).

Por tanto,  $H$  es un plano de corte y (2.26) es su corte asociado. □

Para finalizar este capítulo, resaltamos el hecho de que la convergencia finita del método fraccional de Gomory para los problemas lineales enteros mixtos (PLEM), únicamente se puede asegurar asumiendo que la variable  $z$  asociada al valor de la función objetivo es un número entero. Sin embargo, en general no es razonable asumir en este tipo de problemas que  $z \in \mathbb{Z}$ . La demostración para este caso se omite por su extensión y se puede encontrar en Gomory [3].



## Capítulo 3

# Método de ramificación y acotación

### 3.1. Introducción

Sea el problema lineal entero puro en forma estándar

$$\begin{aligned} \text{Máx} \quad & z(x) = c^T x \\ \text{s.a:} \quad & x \in S^0 \end{aligned} \tag{PLE}$$

donde  $S^0 = \{x \in \mathbb{Z}_+^n : Ax = b\}$  es la región de factibilidad del problema. Al igual que en el punto 2.1, asumimos que  $A \in \mathbb{Z}^{m \times n}$  de rango  $m$ ,  $c \in \mathbb{Z}^n$ ,  $b \in \mathbb{Z}^m$ ,  $m < n$ . Consideramos su problema relajado asociado

$$\begin{aligned} \text{Máx} \quad & z_R(x) = c^T x \\ \text{s.a:} \quad & x \in S_R^0 \end{aligned} \tag{PLE_R}$$

donde  $S_R^0 = \{x \in \mathbb{R}_+^n : Ax = b\}$  es la región de factibilidad del problema relajado. Al igual que en el Capítulo 2, suponemos que  $S_R^0$  está acotado.

Si el número de soluciones factibles del problema (PLE) fuera pequeño, entonces podríamos resolver el problema evaluando cada una de ellas y seleccionaríamos aquella solución factible que proporcionase el máximo valor, en el caso de que  $S^0 \neq \emptyset$ . Este método se conoce como el *Método de enumeración exhaustiva*. Claramente, este método resulta poco práctico cuando el número de soluciones factibles de  $S^0$  es elevado. Por ejemplo, para un problema lineal entero binario con  $n$  variables, el número de soluciones candidatas es de  $2^n$ .

En este capítulo, vamos a mostrar cómo podemos resolver el problema (PLE) empleando el *Método de ramificación y acotación*, el cual se encuentra dentro de la categoría de los conocidos *Métodos de enumeración implícita*. A diferencia del método de enumeración exhaustiva, el método de ramificación y acotación prueba que (PLE) es un problema no factible o, en caso contrario, encuentra la solución óptima haciendo uso únicamente de una *enumeración parcial* del conjunto de soluciones factibles de  $S^0$ .

### 3.2. Algoritmo de ramificación y acotación

El método de ramificación y acotación es uno de los métodos más populares para resolver problemas lineales enteros. Este método se basa en el concepto de *divide y vencerás*, de manera que si no somos capaces de resolver el problema inicial (PLE), podemos dividirlo en subproblemas más pequeños ( $PLE^i$ ) con el objetivo de obtener su solución. Esta idea queda reflejada en la siguiente proposición.

**Proposición 3.1.** Sea la familia  $\{S^i\}_{i=1}^k$  una partición de  $S^0$ . Para todo  $i = 1, \dots, k$ , sean los subproblemas lineales enteros

$$\begin{aligned} \text{Máx} \quad & z^i(x) = c^T x \\ \text{s.a.:} \quad & x \in S^i \end{aligned} \tag{PLE^i}$$

donde  $S^i = \{x \in \mathbb{Z}_+^n : A^i x = b^i\}$  y  $(z^i, x^i)$  es la solución óptima. Si  $(z^0, x^0)$  es la solución óptima de (PLE), entonces

$$z^0 = \max_{1 \leq i \leq k} \{z^i\}.$$

*Demostración.* Para todo  $i = 1, \dots, k$ , se sigue inmediatamente para el subproblema (PLE<sup>i</sup>) que  $z^i \leq z^0$ , ya que  $S^i \subseteq S^0$  y se trata de problemas de máximo. Por otra parte, como la solución óptima de (PLE) verifica que  $x^0 \in S^0$  y  $S^0 = \sqcup_{i=1}^k S^i$ , entonces existe un único  $j$  tal que  $x^0 \in S^j$ , de manera que  $z^j = c^T x^0 = z^0$ . Así pues, se sigue inmediatamente el resultado.  $\square$

El método de ramificación y acotación consiste en realizar una partición (dentro de la literatura del método recibe el nombre de *ramificación*) del conjunto inicial  $S^0$ . De esta manera, por la proposición 3.1, si el problema (PLE) es factible, entonces la solución óptima  $x^0$  será también solución óptima de uno de los subproblemas generados en la partición. En caso contrario, si el problema (PLE) es no factible, entonces los subproblemas generados serán necesariamente no factibles. Suponer que la familia  $\mathcal{F} = \{S^i\}_{i=1}^k$  forma una partición del conjunto inicial  $S^0$ .

**Definición 3.2.** Un conjunto  $S^i \in \mathcal{F}$  se dice que está *sondeado* o que el problema (PLE<sup>i</sup>) se *cierra* si no es necesario realizar ninguna partición sobre el conjunto. Equivalentemente,  $S^i$  está *sondeado* si se verifica alguna de estas tres condiciones:

- (i)  $S^i = \emptyset$ , con lo que (PLE<sup>i</sup>) es no factible.
- (ii) Se ha obtenido la solución óptima  $(z^i, x^i)$  del subproblema (PLE<sup>i</sup>).
- (iii)  $z^i \leq z^0$ .

En general, estamos interesados en establecer cuándo un conjunto  $S^i$  está sondeado sin tener que resolver el problema lineal entero (PLE<sup>i</sup>). Para ello, resolvemos su problema relajado asociado

$$\begin{aligned} \text{Máx} \quad & z_R^i(x) = c^T x \\ \text{s.a.:} \quad & x \in S_R^i \end{aligned} \tag{PLE_R^i}$$

donde  $S_R^i = \{x \in \mathbb{R}_+^n : A^i x = b^i\}$ . De esta manera, podemos extender la definición 3.2 para el caso del problema relajado (PLE<sub>R</sub><sup>i</sup>) asociado de la siguiente manera.

**Proposición 3.3.** Un conjunto  $S^i$  está sondeado si se verifica alguna de estas condiciones:

- (i) (PLE<sub>R</sub><sup>i</sup>) es no factible.
- (ii) La solución óptima  $(z_R^i, x_R^i)$  del problema (PLE<sub>R</sub><sup>i</sup>) verifica que  $x_R^i \in S^i$ .
- (iii)  $z_R^i \leq \underline{z}$ , donde  $\underline{z}$  es el valor para alguna solución factible del problema (PLE).

*Demostración.* Si se sigue (i), entonces (PLE<sub>R</sub><sup>i</sup>) es no factible y se prueba inmediatamente que (PLE<sup>i</sup>) es también no factible. Si se verifica (ii), entonces hemos obtenido la solución óptima del problema (PLE<sup>i</sup>). Finalmente, para el punto (iii), tenemos que se verifican las desigualdades  $z^i \leq z_R^i \leq \underline{z} \leq z^0$ . Para las tres condiciones, es inmediato por la definición 3.2 que el conjunto  $S^i$  está sondeado.  $\square$

A continuación, mostramos el *algoritmo de ramificación y acotación*. En la descripción del algoritmo, denotamos por  $\mathcal{L}$  a la colección de subproblemas enteros lineales  $\{(PLE^i)\}_i$  que todavía no han sido cerrados – el conjunto  $\mathcal{L}$  recibe el nombre de *lista*. Notar además, que cada problema (PLE<sup>i</sup>) tiene asociado una cota superior  $\bar{z}^i$  que será, en nuestro caso, la solución  $z_R^i$  del problema relajado (PLE<sub>R</sub><sup>i</sup>).

Por otra parte, denotaremos por  $\underline{z}$  a la mejor solución encontrada hasta el momento del problema ( $PLE$ ) que, constituirá una cota inferior del valor de la función objetivo  $z^0$ .

El método de ramificación y acotación viene dado por el siguiente algoritmo.

---

**Algoritmo 2** Algoritmo de ramificación y acotación para problemas ( $PLE$ )

---

```

1: Inicializamos:
    $\mathcal{L} \leftarrow \{(PLE^0)\}$  donde  $(PLE^0) \leftarrow (PLE)$ 
    $\bar{z}^0 \leftarrow +\infty, \underline{z} \leftarrow -\infty$ 
2: while 1 do
3:   Paso 1: Test de finalización.
4:   if  $\mathcal{L} = \emptyset$  then
5:     La solución  $x^i$  de  $(PLE^i)$  tal que  $\underline{z} = c^T x^i$  es la solución óptima de  $(PLE)$ . En caso
     de no existir tal  $x^i$ , se sigue que  $(PLE)$  es no factible.
6:     Stop
7:   end if
8:   Paso 2: Selección de subproblema y relajación.
9:   Seleccionamos un subproblema  $(PLE^i) \in \mathcal{L}$  y lo eliminamos de la lista.
10:  Resolvemos su problema relajado asociado  $(PLE_R^i)$  obteniendo la solución óptima
     $(z_R^i, x_R^i)$ , si existe.
11:  Paso 3: Cerrado de subproblemas (conjunto  $S^i$  está sondeado).
12:  if  $z_R^i \leq \underline{z}$  then ir al Paso 1
13:  else if  $x_R^i \in S^i$  (en este caso se sigue que  $z_R^i > \underline{z}$ ) then
14:    Actualizamos  $\underline{z} \leftarrow z_R^i$ 
15:    Eliminamos de  $\mathcal{L}$  todos los subproblemas  $(PLE^i)$  tales que  $\bar{z}^i \leq \underline{z}$ 
16:    ir al Paso 1
17:  else if  $(PLE_R^i)$  es no factible then  $(PLE^i)$  es también no factible.
18:    ir al Paso 1
19:  end if
20:  Paso 4: Ramificación (partición).
21:  if  $x_R^i \notin S^i$  y  $z_R^i > \underline{z}$  then sea  $\{S^{i,j}\}_{j=1}^k$  una partición de  $S^i$ .
22:    Añadimos los subproblemas  $\{(PLE^{i,j})\}_{j=1}^k$  a  $\mathcal{L}$  con  $\bar{z}^{i,j} \leftarrow z_R^i \quad \forall j = 1, \dots, k$ .
23:  end if
24: end while

```

---

El proceso de cerrado de subproblemas indicado en el tercer paso del algoritmo para los casos en los que  $z_R^i \leq \underline{z}$  o  $\bar{z}^i \leq \underline{z}$  recibe el nombre de *cierre por acotación* y, para el caso en el que  $(PLE_R^i)$  es no factible se denomina *cierre por infactibilidad*. Además, observar que las condiciones del tercer paso del algoritmo vienen dadas por la proposición 3.3.

Por otra parte, suponer que resolvemos el problema relajado  $(PLE_R^i)$  mediante el algoritmo simplex dual lexicográfico descrito en el punto 2.3, entonces si en alguna iteración del algoritmo obtenemos que el valor de la SBPFD correspondiente es menor que el valor actual de  $\underline{z}$ , por la proposición 2.6, se seguirá que  $z_R^i \leq \underline{z}$ . Así pues, no es necesario realizar más iteraciones del algoritmo simplex dual lexicográfico para obtener la solución óptima  $z_R^i$  y, podemos aplicar directamente el cierre por acotación de la línea 12 del algoritmo de ramificación y acotación.

A continuación, mostramos la manera en la que se realiza la ramificación de la región de factibilidad  $S^i$ , para todo  $i \geq 0$ .

### 3.3. Ramificación

Sea  $(z_R^i, x_R^i)$  la solución óptima del subproblema  $(PLE_R^i)$  tal que  $x_R^i \notin S^i$  y  $z_R^i > \underline{z}$ , para un  $i \geq 0$ . Así pues, suponer que nos encontramos en el cuarto paso del Algoritmo 2. Una forma de realizar una

partición sobre el conjunto  $S^i$  consiste en añadir restricciones lineales. Para ello, se definen los conjuntos

$$\begin{aligned} S^{i,1} &= S^i \cap \{x \in \mathbb{R}_+^n : d^T x \leq d_0\}, \\ S^{i,2} &= S^i \cap \{x \in \mathbb{R}_+^n : d^T x \geq d_0 + 1\}, \end{aligned} \quad (3.1)$$

donde  $(d_0, d) \in \mathbb{Z}^{n+1}$ . De esta manera, la familia  $\{S^{i,1}, S^{i,2}\}$  forma una partición de  $S^i$ . En efecto, primero denotamos por

$$\begin{aligned} H_1^- &= \{x \in \mathbb{R}_+^n : d^T x \leq d_0\}, \\ H_2^+ &= \{x \in \mathbb{R}_+^n : d^T x \geq d_0 + 1\}, \end{aligned}$$

a los semiespacios cerrados, entonces tenemos que la familia  $\{S^{i,1}, S^{i,2}\}$  verifica:

(i)  $S^{i,1} \cap S^{i,2} = \emptyset$ . Tenemos que

$$S^{i,1} \cap S^{i,2} = S^i \cap (H_1^- \cap H_2^+) = S^i \cap \emptyset = \emptyset,$$

ya que para todo  $x \in H_1^-$ , se sigue que  $d^T x \leq d_0$  y como  $d_0 < d_0 + 1$ , se tiene que  $d^T x < d_0 + 1$ . Así pues,  $x \notin H_2^+$  y  $H_1^- \cap H_2^+ = \emptyset$ .

(ii)  $S^{i,1} \cup S^{i,2} = S^i$ . Tenemos que

$$S^{i,1} \cup S^{i,2} = S^i \cap (H_1^- \cup H_2^+).$$

Por otra parte, notar que

$$H_1^- \cup H_2^+ = \mathbb{R}_+^n \setminus \{x \in \mathbb{R}_+^n : d_0 < d^T x < d_0 + 1\}.$$

Así pues, para todo  $x \in S^i$ , como  $S^i$  es la región de factibilidad del problema  $(PLE^i)$ , se sigue que  $x \in \mathbb{Z}_+^n$  y por tanto,  $x \notin \{x \in \mathbb{R}_+^n : d_0 < d^T x < d_0 + 1\}$  ya que  $(d_0, d) \in \mathbb{Z}^{n+1}$  y  $d^T x \in \mathbb{Z}$ , entonces se tiene necesariamente que  $S^i \subset H_1^- \cup H_2^+$  y, por tanto se sigue el resultado

$$S^{i,1} \cup S^{i,2} = S^i \cap (H_1^- \cup H_2^+) = S^i.$$

De esta manera, hemos probado que la familia  $\{S^{i,1}, S^{i,2}\}$  forma una partición de  $S^i$  para cualquier par  $(d_0, d) \in \mathbb{Z}^{n+1}$ .

El par  $(d_0, d)$  se suele tomar de manera que  $d_0 < d^T x_R^i < d_0 + 1$ . Además, por (3.1) tenemos que

$$\begin{aligned} S_R^{i,1} &= S_R^i \cap \{x \in \mathbb{R}_+^n : d^T x \leq d_0\}, \\ S_R^{i,2} &= S_R^i \cap \{x \in \mathbb{R}_+^n : d^T x \geq d_0 + 1\}, \end{aligned}$$

luego se sigue  $x_R^i \notin S_R^{i,1} \cup S_R^{i,2}$ . Como  $\underline{z} < z_R^i$ , para  $j = 1, 2$ , se puede dar el caso de que la solución óptima  $(z_R^{i,j}, x_R^{i,j})$  correspondiente al subproblema relajado  $(PLE_R^{i,j})$  verifique que  $z_R^{i,j} \leq \underline{z}$  cuando sea  $z_R^{i,j} < z_R^i$ . De esta forma habrá una mayor posibilidad de poder cerrar el subproblema  $(PLE_R^{i,j})$  por acotación.

Para simplificar la notación en los siguientes resultados, denotamos por  $(z^i, x^i)$  a la solución óptima del problema  $(PLE_R^i)$ , de manera que  $(z^i, x^i) = (z_R^i, x_R^i)$ . Como estamos suponiendo que  $x^i \notin S^i$ , entonces existe una variable básica  $x_j^i$  tal que  $x_j^i \notin \mathbb{Z}$  para un  $j \in \{1, \dots, n\}$ . La elección más habitual del par  $(d_0, d)$  es realizada de manera que  $d_0 = \lfloor x_j^i \rfloor$  y  $d = e_j$  verificando que  $\lfloor x_j^i \rfloor < x_j^i < \lfloor x_j^i \rfloor + 1$ , donde  $e_j$  denota al  $j$ -ésimo vector de la base canónica. Esta elección del par  $(d_0, d)$  recibe el nombre de *dicotomía sobre la variable  $x_j$* .

Por otra parte, suponer que hemos resuelto el problema  $(PLE_R^i)$  mediante el algoritmo simplex dual lexicográfico y, que la tabla óptima asociada tiene la forma a la mostrada en la Tabla 2.1, entonces para resolver el subproblema  $(PLE_R^{i,1})$  basta con añadir a la tabla óptima la restricción  $x_j \leq \lfloor x_j^i \rfloor$ . Para ello, añadimos al nuevo problema la variable auxiliar  $s^{i,1} \geq 0$  dada por

$$s^{i,1} = \lfloor x_j^i \rfloor - x_j,$$

y como  $x_j$  es una variable básica, despejando en la tabla óptima, se puede expresar de la forma

$$x_j = x_j^i + \sum_{l \in \Gamma^i} \bar{a}_{j,l}^i (-x_l),$$

para unos coeficientes  $\{\bar{a}_{l,j}^i\}_{l \in \Gamma^i}$  donde  $\Gamma^i$  representa el conjunto de los índices de las variables no básicas de la tabla óptima de  $(PLE_R^i)$ . Por tanto, la variable  $s^{i,1}$  vendrá dada por

$$s^{i,1} = \lfloor x_j^i \rfloor - x_j^i + \sum_{l \in \Gamma^i} (-\bar{a}_{j,l}^i) (-x_l).$$

Siguiendo el mismo razonamiento al realizado en el punto 2.4.1, la tabla aumentada preservará la factibilidad dual. De esta manera, podremos aplicar el algoritmo simplex dual lexicográfico tomando en la primera iteración  $s^{i,1} = \lfloor x_j^i \rfloor - x_j^i < 0$  como variable de salida. Para el subproblema  $(PLE_R^{i,2})$  se procede de manera análoga, añadiendo a la tabla óptima de  $(PLE_R^i)$  la restricción  $x_j \geq \lfloor x_j^i \rfloor + 1$ .

En la práctica, el algoritmo de ramificación y acotación se presenta en forma de árbol, que recibe el nombre de *árbol de enumeración*. El árbol de enumeración está formado por *nodos* que representan cada una de las regiones de factibilidad generadas en las particiones y, por *arcos* donde indicaremos las dicotomías realizadas sobre las variables. Además, para indicar que un nodo está sondeado, colocaremos una barra horizontal situada inmediatamente debajo del nodo correspondiente. Adicionalmente, al lado de los nodos se suele añadir la solución óptima del problema relajado asociado y, el valor de la cota inferior  $\underline{z}$  correspondiente a cada iteración. En la Figura A.5 del Anexo A.2, se puede encontrar un ejemplo de un árbol de enumeración correspondiente al problema propuesto.

Sea  $\lceil \cdot \rceil : \mathbb{R} \rightarrow \mathbb{Z}$  la función *techo entero* dada por

$$\lceil \gamma \rceil = \min\{k \in \mathbb{Z} : \gamma \leq k\},$$

la siguiente proposición nos muestra la finitud de la extensión del árbol de enumeración generado mediante dicotomías sobre las variables.

**Proposición 3.4. Finitud del árbol de enumeración.** Sea  $S_R^0$  la región de factibilidad acotada del problema relajado  $(PLE_R)$ , entonces un árbol de enumeración generado a través de dicotomías sobre las variables es finito. En particular, si  $\omega_j = \lceil \max\{x_j : x \in S_R^0\} \rceil$ , se sigue que ninguna rama del árbol tendrá más de  $\sum_{j=1}^n \omega_j$  arcos.

*Demostración.* Sea  $x^0$  la solución óptima de  $(PLE_R)$  tal que  $x^0 \notin \mathbb{Z}^n$ , entonces existe una variable básica  $x_j^0$  que no toma valores enteros para un  $j \in \{1, \dots, n\}$  y, es claro que  $x_j^0 \leq \max\{x_j : x \in S_R^0\}$ . Por tanto,

$$\lceil x_j^0 \rceil \leq \lceil \max\{x_j^0 : x^0 \in S_R\} \rceil = \omega_j. \quad (3.2)$$

Por otra parte, como  $x_j^0 \notin \mathbb{Z}$  entonces se sigue que  $\lfloor x_j^0 \rfloor = \lfloor x_j^0 \rfloor + 1$  y, sustituyendo en (3.2), tenemos que

$$\lfloor x_j^0 \rfloor \leq \omega_j - 1.$$

Además, como  $x_j^0 \geq 0$ , se sigue finalmente que

$$0 \leq \lfloor x_j^0 \rfloor \leq \omega_j - 1. \quad (3.3)$$

Ahora realizamos una partición del conjunto  $S^0$  mediante una dicotomía sobre la variable  $x_j$ . Por (3.3), añadimos la restricción  $x_j \leq d_1$  para un  $d_1 \in \{0, \dots, \omega_j - 1\}$  y, la restricción  $x_j \geq d_2$  para un  $d_2 \in \{1, \dots, \omega_j\}$  ya que  $d_2 = d_1 + 1$ . Siguiendo la rama de la primera restricción, el resto de dicotomías que se puedan realizar sobre la variable  $x_j$  serán de la forma  $x_j \leq d'_1$  para un  $d'_1 \in \{0, \dots, d_1 - 1\}$  y,  $x_j \geq d''_1$  para un  $d''_1 \in \{1, \dots, d_1\}$ . Análogamente, siguiendo la rama de la segunda restricción, las dicotomías que se puedan realizar sobre la variable  $x_j$  serán de la forma  $x_j \leq d'_2$  para un  $d'_2 \in \{d_2, \dots, \omega_j - 1\}$  y,  $x_j \geq d''_2$  para

un  $d'' \in \{d_2 + 1, \dots, \omega_j\}$ . Por tanto, el mayor número de dicotomías para ambas ramas sobre la variable  $x_j$  ocurrirá cuando añadamos las restricciones  $x_j \leq d \quad \forall d \in \{\omega_j - 1, \dots, 0\}$  o  $x_j \geq d \quad \forall d \in \{1, \dots, \omega_j\}$  o, una combinación  $x_j \geq d \quad \forall d \in \{1, \dots, \alpha\}$  y  $x_j \leq d \quad \forall d \in \{\alpha, \dots, \omega_j - 1\}$ . Es decir, para las tres posibilidades tenemos que el número de restricciones sobre la variable  $x_j$  es de  $\omega_j$ . Por tanto, siguiendo el mismo razonamiento para el resto de variables, cualquier rama del árbol de enumeración tendrá como máximo  $\sum_{j=1}^n \omega_j$  restricciones (arcos).  $\square$

### 3.4. Criterios de selección para el algoritmo de ramificación y acotación

Dedicamos esta sección para abordar los diferentes criterios que se utilizan en la selección de las variables de ramificación por dicotomía y subproblemas correspondientes a los pasos 4 y 2, respectivamente, del método de ramificación y acotación. En la práctica, se ha comprobado la importancia de ambas elecciones para la eficiencia del algoritmo. Por otra parte, por experiencia, la solución óptima  $(z^0, x^0)$  del problema  $(PLE)$  es más probable poder encontrarla en nodos más profundos del árbol de enumeración. Así pues, otro elemento clave para la eficiencia del algoritmo es la capacidad de producir cotas inferiores  $\underline{z}$  próximas al valor de la función objetivo  $z^0$ . De esta manera, habrá una mayor posibilidad de cerrar más subproblemas por acotación y así, poder encontrar la solución óptima  $x^0$  examinando una menor fracción del conjunto de soluciones factibles de  $S^0$ .

#### 3.4.1. Criterio de selección de la variable para ramificar

Hasta ahora, para un subproblema  $(PLE^i)$ , hemos mostrado en el punto 3.3 el procedimiento de ramificación mediante el uso de una dicotomía sobre una variable básica  $x_j^i$  que no toma valores enteros. En general, el número de estas variables no enteras es pequeño. Cuatro de los más habituales criterios de selección de la variable para ramificar son:

- (i) Arbitrariamente, cualquiera de las variables de la base que no toman valores enteros.
- (ii) La variable básica con mayor valor fraccionario.
- (iii) Se asigna prioridades a las variables que deben tomar valores enteros de acuerdo a algún criterio y se elige la variable básica con valor no entero de mayor prioridad.
- (iv) Se elige la variable en la que se alcanza el máximo siguiente:

$$\max_{j \in N^i} \{ \min \{ x_j^i - \lfloor x_j^i \rfloor, \lceil x_j^i \rceil - x_j^i \} \},$$

donde  $N^i = \{j \in N : x_j^i \notin \mathbb{Z}\}$  y  $N = \{1, \dots, n\}$ .

#### 3.4.2. Criterio de selección de los subproblemas

Una vez que nos encontramos en el segundo paso del Algoritmo 2, cuatro posibles criterios de selección del subproblema  $(PLE^i) \in \mathcal{L}$  son:

- (i) Arbitrariamente, cualquier problema de la lista  $\mathcal{L}$ .
- (ii) Se elige el problema en el que se alcanza el siguiente máximo,  $\max_{i \in L} \{z_R^i\}$ , donde  $L = \{i : (PLE^i) \in \mathcal{L}\}$ . De esta forma, se espera que haya una mayor posibilidad de cerrar problemas por acotación. Además, también existirá una mayor posibilidad de encontrar soluciones con mejores valores de la función objetivo.
- (iii) *Last Input First Output (LIFO)*, se toma el subproblema no cerrado más nuevo.
- (iv) *First Input First Output (FIFO)*, se toma el subproblema no cerrado más antiguo.

Para finalizar este capítulo, resaltamos el hecho de que el método de ramificación y acotación se extiende directamente para la resolución de problemas lineales enteros mixtos y, su aplicación es análoga a la explicada para los problemas lineales enteros.

## Capítulo 4

# Aplicación a un problema de planificación de turnos hospitalarios

El Servicio de Urgencias Hospitalarias se define, tomando el diccionario de la RAE, como "la sección de los hospitales en que se atiende a los enfermos y heridos graves que necesitan cuidados médicos inmediatos". El Servicio de Urgencias (SU en adelante) está abierto las 24 horas, todos los días del año. Los médicos deben de trabajar diferentes tipos de turnos tanto de día, de noche y días festivos, sujetos al cumplimiento de un conjunto de restricciones impuestas por Ley. En este capítulo se modeliza el problema de turnos hospitalarios en un SU para el año 2020, con el objetivo de que los turnos asignados sean lo más equilibrados posibles. A continuación, describimos el problema de forma general.

Tenemos  $N$  médicos que están agrupados en  $P$  tipos con  $n_r$  médicos en cada grupo  $G_r$ ,  $r = 1, \dots, P$ . Cada tipo de médico trabaja un número máximo de  $h_r$  horas al año. Por otra parte, en el SU se distingue  $L$  grupos de turnos  $S_j$ ,  $j = 1, \dots, L$  con una duración de  $d_j$  horas para cada turno de tipo  $j$ . Cada médico de grupo  $G_r$  debe de realizar, teóricamente,  $TA_{r,j}$  turnos de  $S_j$  al año. La descripción de los diferentes grupos de médicos y turnos se muestra en las siguientes tablas.

Grupos de médicos	Descripción
$G_1$	Médicos que no realizan turnos de 19/20 horas.
$G_2$	Médicos que realizan el número de turnos de 19/20 horas estándar.
$G_3$	Médicos que realizan dos turnos de 19/20 horas extra al mes.
$G_4$	Médicos que realizan un turno de 19/20 horas extra al mes.
$G_5$	Médicos que realizan medio turno de 19/20 horas extra al mes.
$G_6$	Jefe tipo 1.
$G_7$	Jefe tipo 2.

Tabla 4.1: Descripción de los grupos de médicos  $G_r$ ,  $r = 1, \dots, 7$ .

Grupos de turnos	Descripción
$S_1$	Turnos de 8 horas por el día.
$S_2$	Turnos de 14 horas por el día.
$S_3$	Turnos de 19/20 horas por la noche.
$S_4$	Reunión.

Tabla 4.2: Descripción de los grupos de turnos  $S_j$ ,  $j = 1, \dots, 4$ .

En el año 2020, se distinguen cuatro grupos de días  $T_w$ , caracterizados por el subíndice  $w$  que define el tipo de día y, toma valores en el conjunto  $F = \{0, 1, 3, 5\}$ . Adicionalmente, dentro del grupo  $T_1$ , se diferencian cuatro subgrupos de días  $T_1^k$ , donde  $k \in K = \{WED, WEN, HOLD, HOLN\}$ .

Grupos de días	Descripción
$T_0$	Días laborables, se corresponden a los días: Lunes, Martes, Miércoles sin reunión y Viernes.
$T_1$	Sábados, Domingos y días festivos.
$T_3$	Miércoles con reunión.
$T_5$	Jueves.
$T_1^{WED}$	Fines de semana por el día.
$T_1^{WEN}$	Fines de semana por la noche.
$T_1^{HOLD}$	Días festivos por el día.
$T_1^{HOLN}$	Días festivos por la noche.

Tabla 4.3: Descripción de los grupos de días  $T_w$ ,  $w \in F$  y de  $T_1^k$ ,  $k \in K$ .

Por otra parte, tenemos que cada médico de grupo  $G_r$  debe de realizar, teóricamente,  $TDE_{r,k}$  turnos al año en los días  $t \in T_1^k$ . Además, en el SU se deben de asignar  $ATD_{w,j}$  turnos de  $S_j$  en cada día  $t \in T_w$ .

El número de días correspondientes al año 2020 es de  $T = 366$ , para determinar qué médico se asigna en cada turno cada día a lo largo del año, definimos la variable decisión binaria  $X_{i,j,t}$  dada por

$$X_{i,j,t} = \begin{cases} 1, & \text{si un médico } M_i \text{ trabaja un turno de } S_j \text{ el día } t, \\ 0, & \text{en caso contrario.} \end{cases} \quad \forall i = 1, \dots, N; j = 1, \dots, L; t = 1, \dots, T.$$

El objetivo del problema es obtener una solución factible justa – es decir, ser capaces de obtener una planificación de turnos equitativa, de manera que todos los médicos de un grupo  $G_r$  realicen el mismo número de turnos a lo largo de todos los días festivos, fines de semana y en el año completo. Las restricciones de este problema se categorizan en: (1) *restricciones fuertes*, las cuales se deben de satisfacer por Ley y, (2) *restricciones suaves*, que se pueden incumplir sujetas a unas penalizaciones en la función objetivo. Los datos teóricos  $TA_{r,j}$  y  $TDE_{r,k}$  garantizan un reparto equilibrado de los turnos y, por tanto, las restricciones suaves se emplean para aproximarnos lo máximo posible a estos valores.

La planificación de turnos hospitalarios en un SU se formula como un problema de PLE binario mixto, donde las variables de decisión binarias son las variables  $X_{i,j,t}$  y, las variables reales se corresponden a las llamadas *variables metas* que nos permitirán modelar las restricciones suaves. De esta forma, el problema minimiza la suma de todos los valores de las variables metas, garantizando que la asignación de las variables binarias es justa y verifican las restricciones fuertes. Los datos numéricos del problema se encuentran en el Anexo B.2 y, su formulación matemática se presenta en el Anexo B.3.

El modelo de programación consta de 63640 variables de decisión y 51208 restricciones. El problema se resuelve mediante el lenguaje de modelado CPLEX versión 12.90 desarrollado por IBM, en un Intel(R) Core(TM) i7-3537U CPU @ 2.00 GHz con 8.00 GB de RAM. El tiempo de ejecución es de 9 minutos 48 segundos y, el valor de la solución objetivo es de 103.71 (turnos). Debido a las grandes dimensiones del problema, las variables de decisión  $X_{i,j,t}$  se exportan a un libro de Excel a una matriz de dimensión  $43 \times 366$  – en la Tabla B.10 se muestran las asignaciones obtenidas para el mes de Enero. Por otra parte, el conjunto de las variables metas representa el indicador sobre la equidad en la asignación y no, el valor global de la función objetivo. Por ello, en la misma hoja de Excel, se exporta los valores de las variables metas. En la Tabla B.9, se observa que todos los valores son menores que 1, lo cual nos indica que el reparto de los turnos es muy próximo a los valores teóricos y, por tanto, la planificación es justa. En el Anexo B.4 se muestra el código de resolución del problema en CPLEX.



# Conclusiones

A continuación, se exponen las conclusiones obtenidas del presente trabajo para cada uno de los capítulos desarrollados:

- **Introducción a la programación lineal entera.** Se ha presentado brevemente los orígenes de la PL para introducir la necesidad y motivación del desarrollo de la PLE. A continuación, se ha mostrado la gran utilidad de su aplicación con dos casos reales, la amplia gama de problemas que permite formular y el motivo del aumento de la complejidad en su resolución. Finalmente, como introducción a los métodos que se desarrollan, se ha mostrado los métodos de redondeo que permiten – para un número reducido de problemas – una resolución sencilla y directa.
- **Método de planos de corte.** Tras definir las nociones de planos de corte, se ha presentado el algoritmo del Simplex dual lexicográfico, que es la base del desarrollo del método fraccional de Gomory. Posteriormente, se ha mostrado el algoritmo fraccional de Gomory para PLE derivando el corte fraccional de Gomory y, se han presentado una serie de proposiciones que nos han permitido probar la convergencia finita del método. Finalmente, se ha desarrollado la formulación del corte fraccional de Gomory para el caso de PLEM.
- **Método de ramificación y acotación.** En la práctica, es infrecuente el uso del método de Gomory. Por ello, en este capítulo se ha desarrollado el método de ramificación y acotación que resulta más eficiente. Primero, se ha mostrado el algoritmo de ramificación y acotación para la resolución de PLE. Posteriormente, se ha desarrollado el método de la dicotomía sobre la variable, permitiéndonos probar la finitud del árbol de enumeración. Finalmente, hemos indicado cuatro de los criterios más habituales en la selección de la variable para ramificar y elección de los subproblemas.

Adicionalmente, en el Anexo A, hemos mostrado un problema de máximo para ejemplificar la resolución práctica de estos dos métodos.

- **Aplicación a un problema de planificación de turnos hospitalarios.** Se ha presentado la descripción del problema de forma general, definiendo los parámetros y variables necesarias para su modelización. Posteriormente, indicando brevemente las restricciones y el objetivo del problema, se han expuesto los resultados obtenidos. El Anexo B, se ha presentado como extensión del capítulo, en donde hemos mostrado la notación, los datos y la formulación matemática del problema – definiendo el conjunto de las restricciones fuertes y suaves. Por otra parte, se ha recogido el código de resolución en CPLEX y, por último, se han presentado los resultados en las variables metas y la asignación obtenida para el mes de Enero de 2020.

Finalmente, en este trabajo se ha desarrollado un método exacto para la resolución del problema obteniendo un tiempo de ejecución razonable. Sin embargo, en la práctica, se considera un mayor número de restricciones y, dentro de cada tipo de turno de  $S_j$ , se diferencia otro subconjunto de turnos. Con este aumento en la complejidad, resulta inviable – por tiempo de ejecución – resolverlo de forma exacta. En la actualidad, existen líneas de investigación que abordan este problema con el uso de métodos heurísticos que permiten obtener una asignación justa en un tiempo reducido.



# Bibliografía

- [1] COOK, W. J. (2012). Markowitz and Manne + Eastman + Land and Doig = Branch and Bound. *Documenta Mathematica, Extra Volume: Optimization Stories*, 227–238.
- [2] GOMORY, R. E. An Algorithm for Integer Solutions to Linear Programs. En GRAVES, R. L., & WOLFE, P. (1963). *Recent Advances in Mathematical Programming* (pp. 269–302). New York: McGrawHill.
- [3] GOMORY, R. E. (1960). An Algorithm for the Mixed Integer Problem. *Rand Report*, RM-2597-PR. Santa Monica, CA: Rand Corporation.
- [4] GOMORY, R. E. (1958). Outline of an Algorithm for Integer Solutions to Linear Programs. *Bulletin of the American Mathematical Society*, 64(5), 275–278.
- [5] HILLIER, F. S., Y LIEBERMAN, G. J. (2010). *Introducción a la Investigación de Operaciones* (9.<sup>a</sup> ed.). México: McGrawHill.
- [6] MALLOR, F., MATEO, P. M., & CILDOZ, M. (2020). A Hybrid Grasp-Based Algorithm to Solve the Emergency Room Physician Scheduling Problem. A Case Study. Se encuentra en revisión en *Applied Soft Computing*.
- [7] MURTY, K. G. (1976). *Linear and Combinatorial Programming*. New York: John Wiley & Sons.
- [8] NEMHAUSER, G. L., & WOLSEY, L. A. (1988). *Integer and Combinatorial Optimization*. New York: John Wiley & Sons.



# Anexo A. Ejemplos de resolución aplicando los métodos propuestos

El objeto de este anexo es el de mostrar la aplicación directa del método fraccional de Gomory y, del método de ramificación y acotación a la hora de resolver problemas lineales enteros. Para ello, proponemos el siguiente problema de máximo tomado de Nemhauser & Wolsey [8, sección II.4.3].

Sea el problema lineal entero puro

$$\begin{aligned} \text{Máx} \quad & z(x_1, x_2) = 7x_1 + 2x_2 \\ \text{s.a:} \quad & -x_1 + 2x_2 \leq 4 \\ & 5x_1 + x_2 \leq 20 \\ & x_1, x_2 \in \mathbb{Z}^+ \end{aligned} \tag{PLE}$$

Introducimos las variables de holgura  $x_3, x_4 \in \mathbb{Z}^+$  para expresar el problema en forma estándar resultando

$$\begin{aligned} \text{Máx} \quad & z(x_1, x_2) = 7x_1 + 2x_2 \\ \text{s.a:} \quad & -x_1 + 2x_2 + x_3 = 4 \\ & 5x_1 + x_2 + x_4 = 20 \\ & x_1, x_2, x_3, x_4 \in \mathbb{Z}^+ \end{aligned}$$

En los siguientes apartados se muestra la resolución detallada de este problema mediante el uso de estos dos métodos. En Nemhauser et al. [8, sección II.4.3] únicamente se muestran las soluciones de las iteraciones resultantes al aplicar el algoritmo de Gomory.

## A.1. Resolución con el método fraccional de Gomory

Resolvemos este problema lineal entero en forma estándar aplicando el método fraccional de Gomory mostrado en el Capítulo 2. En cada iteración del algoritmo, haremos uso de la notación empleada en el punto 2.3 para denotar a los elementos de las tablas resultantes de cada iteración de los algoritmos simplex primal y simplex dual lexicográfico, respectivamente. En la primera iteración del método fraccional de Gomory haremos uso del algoritmo simplex primal y, en las siguientes iteraciones aplicaremos el algoritmo simplex dual lexicográfico para resolver cada uno de los problemas relajados.

### Iteración 1

Resolvemos el problema relajado  $(PLE_R^1)$ . La tabla inicial viene dada por

		$-x_1$	$-x_2$
$z$	0	-7	-2
$x_3$	4	-1	2
$x_4$	20	5	1
$x_1$	0	-1	0
$x_2$	0	0	-1

Esta tabla constituye una solución factible básica (SFB en adelante) ya que  $\bar{b}_i^0 \geq 0 \quad \forall i = 1, \dots, 4$ , pero la solución no es óptima al tener los costos marginales (cambiados de signo) asociados a las variables básicas negativos, esto es,  $\bar{c}_1^0 = -7 < 0$ ,  $\bar{c}_2^0 = -2 < 0$ . Aplicamos el algoritmo simplex primal utilizando los criterios de selección de las variables de entrada y salida de la base que definen este algoritmo. De esta forma, elegimos la variable  $x_1$  para entrar en la base ya que se corresponde a la columna donde se alcanza el siguiente mínimo,  $\min\{-7, -2\} = -7$ . Por otra parte, elegimos la variable  $x_4$  para salir de la base porque es la variable básica donde se alcanza el siguiente mínimo:

$$\min_{i \in \{3,4\}} \left\{ \frac{\bar{b}_i^0}{\bar{a}_{i,1}^0} : \bar{a}_{i,1}^0 > 0 \right\} = \min\{4\} = 4.$$

Aplicamos las operaciones de pivotaje correspondientes a esta tabla dadas por:

$$\begin{aligned} D_4^1 &= -\frac{1}{5}D_1^0, \\ D_0^1 &= D_0^0 - 4D_1^0, \\ D_2^1 &= D_2^0 - \frac{1}{5}D_1^0. \end{aligned}$$

De esta manera, la tabla resultante es

		$-x_4$	$-x_2$
$z$	28	$\frac{7}{5}$	$-\frac{3}{5}$
$x_3$	8	$\frac{1}{5}$	$\frac{11}{5}$
$x_4$	0	-1	0
$x_1$	4	$\frac{1}{5}$	$\frac{1}{5}$
$x_2$	0	0	-1

Esta tabla constituye una SFB pero no es óptima al ser  $c_2^1 = -\frac{3}{5} < 0$ . Por tanto, aplicamos el algoritmo simplex primal tomando  $x_2$  como variable de entrada en la base y,  $x_3$  como variable de salida ya que se corresponde a la fila en donde se alcanza el siguiente mínimo:

$$\min_{i \in \{1,3\}} \left\{ \frac{\bar{b}_i^1}{\bar{a}_{i,2}^1} : \bar{a}_{i,2}^1 > 0 \right\} = \min\left\{\frac{40}{11}, 20\right\} = \frac{40}{11}.$$

Realizamos las operaciones de pivotaje asociadas definidas por:

$$\begin{aligned} D_3^2 &= -\frac{5}{11}D_2^1, \\ D_0^2 &= D_0^1 - \frac{40}{11}D_2^1, \\ D_4^2 &= D_4^1 - \frac{1}{11}D_2^1, \end{aligned}$$

obteniendo

		$-x_4$	$-x_3$
$z$	$\frac{332}{11}$	$\frac{16}{11}$	$\frac{3}{11}$
$x_3$	0	0	-1
$x_4$	0	-1	0
$x_1$	$\frac{36}{11}$	$\frac{2}{11}$	$-\frac{1}{11}$
$x_2$	$\frac{40}{11}$	$\frac{1}{11}$	$\frac{5}{11}$

Esta tabla resultante constituye una SFB y es óptima al tener los costos marginales (cambiados de signo)  $\bar{c}_4^2, \bar{c}_3^2 \geq 0$ . Por tanto, hemos obtenido la solución óptima del problema ( $PLE_R^1$ ), sin embargo, esta solución no verifica las condiciones de integridad. Así pues, añadimos el corte de Gomory

$$s^1 = -\frac{2}{11} - \frac{5}{11}(-x_4) - \frac{3}{11}(-x_3)$$

asociado a la variable  $z$ , que corresponde a la variable no entera de menor índice. A continuación, procedemos con la segunda iteración del algoritmo fraccional de Gomory.

### Iteración 2

Añadimos el corte de Gomory a la tabla óptima correspondiente al problema anterior ( $PLE_R^1$ ) obteniendo

		$-x_4$	$-x_3$
$z$	$\frac{332}{11}$	$\frac{16}{11}$	$\frac{3}{11}$
$x_3$	0	0	-1
$x_4$	0	-1	0
$x_1$	$\frac{36}{11}$	$\frac{2}{11}$	$-\frac{1}{11}$
$x_2$	$\frac{40}{11}$	$\frac{1}{11}$	$\frac{5}{11}$
$s^1$	$-\frac{2}{11}$	$-\frac{5}{11}$	$-\frac{3}{11}$

Notar que esta tabla no constituye una SFB para el problema ( $PLE_R^2$ ) ya que  $s^1 = \frac{-2}{11} < 0$ . Por tanto, no podemos aplicar el algoritmo simplex primal, sin embargo, esta tabla constituye una SBPFD para el problema ( $PLE_R^2$ ) al tener los coeficientes  $\bar{c}_4^0, \bar{c}_3^0$  no negativos. A continuación, aplicamos el algoritmo simplex dual lexicográfico. Elegimos la variable  $s^1$  para abandonar la base y seleccionamos la variable  $x_3$  para entrar en la base, siguiendo los criterios de selección proporcionados en las proposiciones 2.6 y 2.7, respectivamente. Las operaciones de pivotaje asociadas vienen dadas por:

$$\begin{aligned} D_{s^1}^1 &= \frac{11}{3} D_3^0, \\ D_0^1 &= D_0^0 - \frac{2}{3} D_3^0, \\ D_4^1 &= D_4^0 - \frac{5}{3} D_3^0, \end{aligned}$$

obteniendo la tabla

		$-x_4$	$-s^1$
$z$	30	1	1
$x_3$	$\frac{2}{3}$	$\frac{5}{3}$	$-\frac{11}{3}$
$x_4$	0	-1	0
$x_1$	$\frac{10}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$
$x_2$	$\frac{10}{3}$	$-\frac{2}{3}$	$\frac{5}{3}$
$s^1$	0	0	-1

La tabla resultante es una SBPFD y es factible primal. Por tanto, es óptima para el problema ( $PLE_R^2$ ). De nuevo, la solución resultante no verifica las condiciones de integridad, con lo que añadimos el corte de Gomory

$$s^2 = -\frac{2}{3} - \frac{2}{3}(-x_4) - \frac{1}{3}(-s^1)$$

asociado a la variable  $x_3$ .

### Iteración 3

La tabla ampliada resultante al añadir el corte de Gomory es

		$-x_4$	$-s^1$
$z$	30	1	1
$x_3$	$\frac{2}{3}$	$\frac{5}{3}$	$-\frac{11}{3}$
$x_4$	0	-1	0
$x_1$	$\frac{10}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$
$x_2$	$\frac{10}{3}$	$-\frac{2}{3}$	$\frac{5}{3}$
$s^1$	0	0	-1
$s^2$	$-\frac{2}{3}$	$-\frac{2}{3}$	$-\frac{1}{3}$

Esta tabla constituye una SBPFD para el problema  $(PLE_R^3)$ . Aplicamos el algoritmo simplex dual lexicográfico al no constituir una solución factible primal. Al igual que antes, aplicando los criterios de selección de las variables de salida y entrada en la base tenemos:

Sale  $s^2$ , entra  $x_4$ .

		$-s^2$	$-s^1$
$z$	29	$\frac{3}{2}$	$\frac{1}{2}$
$x_3$	-1	$\frac{5}{2}$	$-\frac{9}{2}$
$x_4$	1	$-\frac{3}{2}$	$\frac{1}{2}$
$x_1$	3	$\frac{1}{2}$	$-\frac{1}{2}$
$x_2$	4	-1	2
$s^1$	0	0	-1
$s^2$	0	-1	0

Sale  $x_3$ , entra  $s^1$ .

		$-s^2$	$-x_3$
$z$	$\frac{260}{9}$	$\frac{16}{9}$	$\frac{1}{9}$
$x_3$	0	0	-1
$x_4$	$\frac{8}{9}$	$-\frac{11}{9}$	$\frac{1}{9}$
$x_1$	$\frac{28}{9}$	$\frac{2}{9}$	$-\frac{1}{9}$
$x_2$	$\frac{32}{9}$	$\frac{1}{9}$	$\frac{4}{9}$
$s^1$	$\frac{2}{9}$	$-\frac{5}{9}$	$-\frac{2}{9}$
$s^2$	0	-1	0

La tabla resultante es factible primal y, por tanto, óptima para el problema  $(PLE_R^3)$  pero no verifica las condiciones de integridad. Así pues, añadimos el corte de Gomory

$$s^3 = -\frac{8}{9} - \frac{7}{9}(-s^2) - \frac{1}{9}(-x_3)$$



asociado a la variable  $z$ . Notar que la variable  $s^1$  toma valor positivo en la base y, por (2.11), podemos eliminar su fila asociada ya que el corte está desactivado.

#### Iteración 4

La tabla ampliada resultante al añadir el corte de Gomory y eliminar la fila asociada a la variable  $s^1$  es

		$-s^2$	$-x_3$
$z$	$\frac{260}{9}$	$\frac{16}{9}$	$\frac{1}{9}$
$x_3$	0	0	-1
$x_4$	$\frac{8}{9}$	$-\frac{11}{9}$	$\frac{1}{9}$
$x_1$	$\frac{28}{9}$	$\frac{2}{9}$	$-\frac{1}{9}$
$x_2$	$\frac{32}{9}$	$\frac{1}{9}$	$\frac{4}{9}$
$s^2$	0	-1	0
$s^3$	$-\frac{8}{9}$	$-\frac{7}{9}$	$-\frac{1}{9}$

formando una SBPFD para el problema  $(PLE_R^4)$  pero no es una solución factible primal. Aplicamos el algoritmo simplex dual lexicográfico.

Sale  $s^3$ , entra  $x_3$ .

		$-s^2$	$-s^3$
$z$	28	1	1
$x_3$	8	7	-9
$x_4$	0	-2	1
$x_1$	4	1	-1
$x_2$	0	-3	4
$s^2$	0	-1	0
$s^3$	0	0	-1

La tabla obtenida es óptima para el problema  $(PLE_R^4)$  por ser factible primal. Además, la solución obtenida verifica las condiciones de integridad. Por tanto, hemos sido capaces de resolver el problema de máximo  $(PLE)$  aplicando el algoritmo fraccional de Gomory en cuatro iteraciones. La solución del problema  $(PLE)$  es  $x_1 = 4$ ,  $x_2 = 0$  con un valor de la función objetivo de 28.

## A.2. Resolución con el método de ramificación y acotación

A continuación, resolvemos el problema lineal entero  $(PLE)$  en forma estándar aplicando el método de ramificación y acotación mostrado en el Capítulo 3. Para la realización de este método, utilizamos el criterio de selección de la variable para ramificar mostrado en el punto 3.4.1 apartado (iv) y, el criterio (LIFO) dado en el punto 3.4.2 apartado (iii) para la selección de los subproblemas. En caso de empate en este último criterio, elegiremos el subproblema situado más a la izquierda en el árbol de enumeración. Por otra parte, en los árboles de enumeración obtenidos en cada iteración únicamente mostraremos al lado de los nodos el valor de la cota inferior  $\underline{z}$  correspondiente. En el último árbol, indicaremos además los valores de las soluciones óptimas de los problemas relajados que han sido resueltos.

#### Iteración 1

Empezamos con el conjunto lista  $\mathcal{L} = \{(PLE^0)\}$ . Seleccionamos el problema  $(PLE^0)$  eliminándolo de la lista y, resolvemos el problema relajado  $(PLE_R^0)$ . Este problema ya lo hemos resuelto en el Anexo

A.1 y su tabla óptima es

		$-x_4$	$-x_3$
$z$	$\frac{332}{11}$	$\frac{16}{11}$	$\frac{3}{11}$
$x_3$	0	0	-1
$x_4$	0	-1	0
$x_1$	$\frac{36}{11}$	$\frac{2}{11}$	$-\frac{1}{11}$
$x_2$	$\frac{40}{11}$	$\frac{1}{11}$	$\frac{5}{11}$

Notar que la solución óptima no pertenece al conjunto  $S^0$  al no verificar las condiciones de integridad. Así pues, por el cuarto paso del Algoritmo 2, realizamos una partición de  $S^0$  mediante una dicotomía sobre la variable  $x_2$ , ya que se corresponde con la variable donde se alcanza el siguiente máximo:

$$\max \left\{ \min \left\{ \frac{3}{11}, \frac{8}{11} \right\}, \min \left\{ \frac{7}{11}, \frac{4}{11} \right\} \right\} = \max \left\{ \frac{3}{11}, \frac{4}{11} \right\} = \frac{4}{11}.$$

De esta forma, hemos generado la familia  $\{S^{01}, S^{02}\}$  que constituye una partición de  $S^0$  añadiendo las restricciones  $x_2 \leq \lfloor \frac{40}{11} \rfloor = 3$ ,  $x_2 \geq \lceil \frac{40}{11} \rceil = 4$ , respectivamente. Añadimos los problemas lineales enteros asociados a estos conjuntos a la lista. Así pues, en esta iteración  $\mathcal{L} = \{(PLE^{01}), (PLE^{02})\}$ .

El árbol de enumeración es

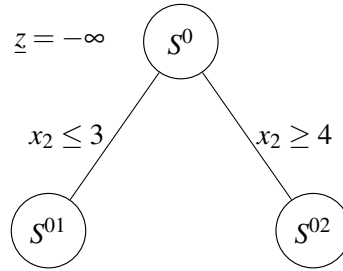


Figura A.1: Árbol de enumeración para la primera iteración.

## Iteración 2

Siguiendo el criterio (LIFO) tomamos el subproblema situado más a la izquierda en el árbol de enumeración, seleccionamos el problema  $(PLE^{01})$  eliminándolo de la lista  $\mathcal{L}$  y resolvemos el problema relajado  $(PLE_R^{01})$ . Para ello, añadimos a la tabla óptima de  $(PLE_R^0)$  la restricción  $x_2 \leq 3$ . Esto es, introduciendo la variable auxiliar  $s^{01} \geq 0$  y despejando  $x_2$  de la tabla óptima, tenemos la restricción

$$s^{01} = -\frac{7}{11} - \frac{1}{11}(-x_4) - \frac{5}{11}(-x_3).$$

De esta forma, la tabla resultante es

		$-x_4$	$-x_3$
$z$	$\frac{332}{11}$	$\frac{16}{11}$	$\frac{3}{11}$
$x_3$	0	0	-1
$x_4$	0	-1	0
$x_1$	$\frac{36}{11}$	$\frac{2}{11}$	$-\frac{1}{11}$
$x_2$	$\frac{40}{11}$	$\frac{1}{11}$	$\frac{5}{11}$
$s^{01}$	$-\frac{7}{11}$	$-\frac{1}{11}$	$-\frac{5}{11}$

Esta tabla constituye una SBPFD para el problema  $(PLE_R^{01})$  pero no es una solución factible primal. Aplicamos el algoritmo simplex dual lexicográfico.

Sale  $s^{01}$  y entra  $x_3$ .

		$-x_4$	$-s^{01}$
$z$	$\frac{149}{5}$	$\frac{7}{5}$	$\frac{3}{5}$
$x_3$	$\frac{7}{5}$	$\frac{1}{5}$	$-\frac{11}{5}$
$x_4$	0	-1	0
$x_1$	$\frac{17}{5}$	$\frac{1}{5}$	$-\frac{1}{5}$
$x_2$	3	0	1
$s^{01}$	0	0	-1

La tabla resultante es factible primal y, por tanto, óptima para el problema  $(PLE_R^{01})$ . Al igual que en la primera iteración, la solución obtenida no verifica las condiciones de integridad. Por tanto, realizamos una partición del conjunto  $S^{01}$  seleccionando la variable donde se alcanza el siguiente máximo:

$$\max \left\{ \min \left\{ \frac{2}{5}, \frac{3}{5} \right\}, \min \left\{ \frac{2}{5}, \frac{3}{5} \right\} \right\} = \max \left\{ \frac{2}{5}, \frac{2}{5} \right\} = \frac{2}{5}.$$

En este caso, observar que el criterio de elección no decide ya que las variables no enteras  $x_3, x_1$  de la tabla alcanzan el máximo anterior. Por tanto, decidimos realizar arbitrariamente una dicotomía sobre la variable  $x_1$ . Generamos la familia  $\{S^{011}, S^{012}\}$  añadiendo las restricciones  $x_1 \leq 3, x_1 \geq 4$ , respectivamente. Añadimos los problemas asociados a la lista obteniendo  $\mathcal{L} = \{(PLE^{02}), (PLE^{011}), (PLE^{012})\}$ .

El árbol de enumeración es

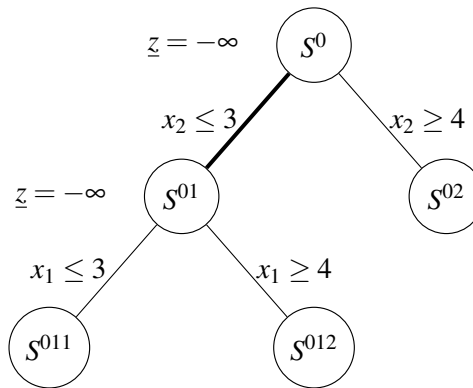


Figura A.2: Árbol de enumeración para la segunda iteración.

### Iteración 3

Seleccionamos el problema  $(PLE^{011})$  y lo eliminamos de la lista  $\mathcal{L}$ . Resolvemos el problema relacionado  $(PLE_R^{011})$  añadiendo la restricción  $x_1 \leq 3$  a la tabla óptima de  $(PLE_R^{01})$ . Para ello, introducimos la variable auxiliar  $s^{011} \geq 0$  y despejando  $x_1$  de la tabla óptima, tenemos la restricción

$$s^{011} = -\frac{2}{5} - \frac{1}{5}(-x_4) + \frac{1}{5}(-s^{01}).$$

La tabla resultante es

		$-x_4$	$-s^{01}$
$z$	$\frac{149}{5}$	$\frac{7}{5}$	$\frac{3}{5}$
$x_3$	$\frac{7}{5}$	$\frac{1}{5}$	$-\frac{11}{5}$
$x_4$	0	-1	0
$x_1$	$\frac{17}{5}$	$\frac{1}{5}$	$-\frac{1}{5}$
$x_2$	3	0	1
$s^{01}$	0	0	-1
$s^{011}$	$-\frac{2}{5}$	$-\frac{1}{5}$	$\frac{1}{5}$

Esta tabla constituye una SBPFD para el problema  $(PLE_R^{011})$ . Aplicamos el algoritmo simplex dual lexicográfico al no constituir una solución factible primal.

Sale  $s^{011}$ , entra  $x_4$ .

		$-s^{011}$	$-s^{01}$
$z$	27	7	2
$x_3$	1	1	-2
$x_4$	2	-5	-1
$x_1$	3	1	0
$x_2$	3	0	1
$s^{01}$	0	0	-1
$s^{011}$	0	-1	0

La tabla es factible primal y, por tanto, es óptima para el problema  $(PLE_R^{011})$ . Además, la solución  $x_R^{011} \in S^{011}$  por verificar las condiciones de integridad. Por otra parte,  $z_R^{011} = 27 > \underline{z} = -\infty$ , con lo que actualizamos el valor de la cota inferior a  $\underline{z} = 27$  y, cerramos el problema  $(PLE^{011})$  por acotación ya que por debajo no habrá ninguna solución mejor que la recién alcanzada. En esta iteración, tenemos la lista  $\mathcal{L} = \{(PLE^{02}), (PLE^{012})\}$ .

El árbol de enumeración es

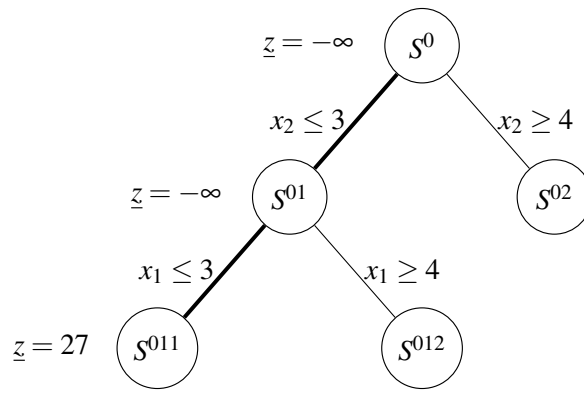


Figura A.3: Árbol de enumeración para la tercera iteración.

#### Iteración 4

Seleccionamos el problema  $(PLE^{012})$  y lo eliminamos de  $\mathcal{L}$ . Resolvemos el problema relajado  $(PLE_R^{012})$  añadiendo la restricción  $x_1 \geq 4$  a la tabla óptima de  $(PLE_R^{01})$ . Introduciendo la variable auxiliar

$s^{012} \geq 0$  y despejando  $x_1$  de la tabla óptima, obtenemos la restricción

$$s^{012} = -\frac{3}{5} + \frac{1}{5}(-x_4) - \frac{1}{5}(-s^{01}).$$

La tabla resultantes es

		$-x_4$	$-s^{01}$
$z$	$\frac{149}{5}$	$\frac{7}{5}$	$\frac{3}{5}$
$x_3$	$\frac{7}{5}$	$\frac{1}{5}$	$-\frac{11}{5}$
$x_4$	0	-1	0
$x_1$	$\frac{17}{5}$	$\frac{1}{5}$	$-\frac{1}{5}$
$x_2$	3	0	1
$s^{01}$	0	0	-1
$s^{012}$	$-\frac{3}{5}$	$\frac{1}{5}$	$-\frac{1}{5}$

Esta tabla constituye un SBPFD para el problema  $(PLE_R^{012})$  pero es no factible primal. Aplicamos el algoritmo simplex dual lexicográfico.

Sale  $s^{012}$ , entra  $s^{01}$ .

		$-x_4$	$-s^{011}$
$z$	28	2	3
$x_3$	8	-2	-11
$x_4$	0	-1	0
$x_1$	4	0	-1
$x_2$	0	1	5
$s^{01}$	3	-1	-5
$s^{012}$	0	0	-1

La tabla resultante es óptima para el problema  $(PLE_R^{012})$  por ser factible primal. Además, la solución verifica las condiciones de integridad, con lo que se tiene que la solución  $x_R^{012} \in S^{012}$ . Por otra parte, como se sigue que  $z_R^{012} = 28 > \underline{z} = 27$ , actualizamos el valor de la cota inferior a  $z = 28$  y cerramos el problema  $(PLE^{012})$  por acotación. En esta iteración, tenemos la lista  $\mathcal{L} = \{(PLE^{02})\}$ .

El árbol de enumeración es

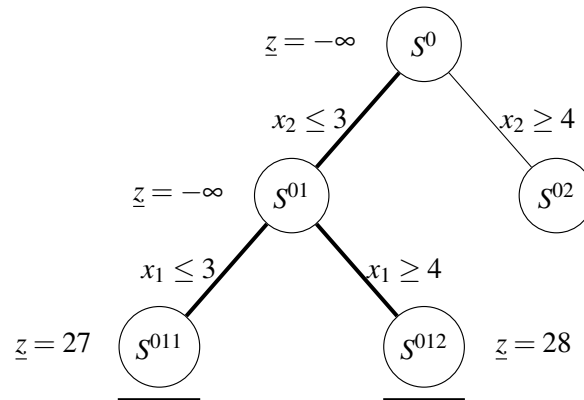


Figura A.4: Árbol de enumeración para la cuarta iteración.

**Iteración 5**

Seleccionamos el problema  $(PLE^{02})$  y lo eliminamos de la lista. Resolvemos el problema relajado  $(PLE_R^{02})$  añadiendo la restricción  $x_2 \geq 4$  a la tabla óptima de  $(PLE_R^0)$ . Para ello, introducimos la variable auxiliar  $s^{02} \geq 0$  y despejamos  $x_2$  de la tabla óptima, obteniendo la restricción

$$s^{02} = -\frac{4}{11} + \frac{1}{11}(-x_4) + \frac{5}{11}(-x_3).$$

La tabla resultante es

		$-x_4$	$-x_3$
$z$	$\frac{332}{11}$	$\frac{16}{11}$	$\frac{3}{11}$
$x_3$	0	0	-1
$x_4$	0	-1	0
$x_1$	$\frac{36}{11}$	$\frac{2}{11}$	$-\frac{1}{11}$
$x_2$	$\frac{40}{11}$	$\frac{1}{11}$	$\frac{5}{11}$
$s^{02}$	$-\frac{4}{11}$	$\frac{1}{11}$	$\frac{5}{11}$

Esta tabla constituye una SBPFD para el problema  $(PLE_R^{02})$  pero es no factible primal. Aplicamos el algoritmo simplex dual lexicográfico.

Elegimos la variable  $s^{02}$  para abandonar la base, pero los elementos  $\frac{1}{11}$ ,  $\frac{5}{11}$  de la fila correspondiente a esta variable son no negativos, entonces por la proposición 2.7 el problema  $(PLE_R^{02})$  es no factible, con lo que  $(PLE^{02})$  es también no factible. Por tanto, cerramos el problema  $(PLE^{02})$  por infactibilidad.

Finalmente, el árbol de enumeración del algoritmo añadiendo las soluciones óptimas correspondientes a cada problema relajado es

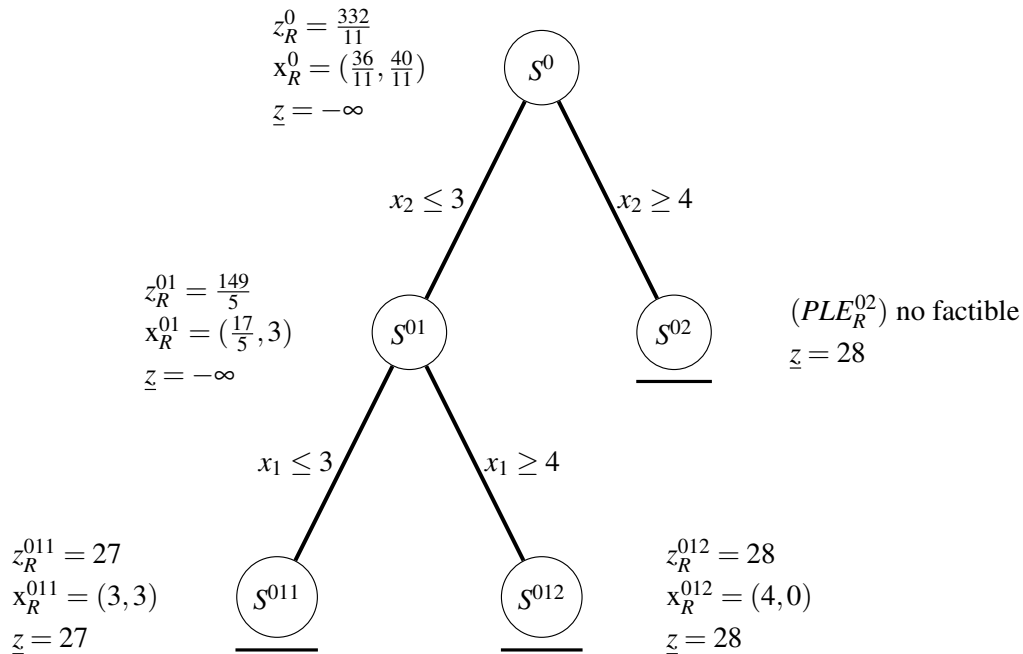


Figura A.5: Árbol de enumeración completo del problema  $(PLE)$ .

**Iteración 6**

Tenemos que  $\mathcal{L} = \emptyset$  y como la solución  $x^{012} = (4, 0)$  del problema  $(PLE^{012})$  proporciona el valor actual de la cota inferior  $\underline{z}$ , tenemos que la solución óptima del problema  $(PLE)$  es  $x_1 = 4$ ,  $x_2 = 0$  con

un valor de la función objetivo de 28. Por tanto, hemos sido capaces de resolver el problema de máximo (*PLE*) realizando seis iteraciones del algoritmo de ramificación y acotación.





# Anexo B. Problema de planificación de turnos hospitalarios

Este anexo se presenta como complemento al Capítulo 4 para: (1) mostrar la notación y los datos del problema que se modela, (2) mostrar la formulación matemática del problema de PLE binario mixto, (3) presentar el código de resolución realizado en CPLEX y, (4) mostrar los resultados obtenidos.

## B.1. Notación

Notación	Definición y dominio
Parámetros	
$N$	Número total de médicos
$M_i$	Un médico $i$ , $i = 1, \dots, N$
$P$	Número de diferentes tipos de médicos
$G_r$	Grupo de médicos de tipo $r$ , $r = 1, \dots, P$
$n_r$	Número de médicos de grupo $G_r$ , $r = 1, \dots, P$
$h_r$	Número de horas totales que un médico de grupo $G_r$ trabaja en el año, $r = 1, \dots, P$
$L$	Número de diferentes tipos de turnos
$S_j$	Grupo de turnos de tipo $j$ , $j = 1, \dots, L$
$d_j$	Número de horas que dura un turno de $S_j$ , $j = 1, \dots, L$
$T$	Número de días en el año 2020
$F$	Conjunto de diferentes tipos de días, donde $F = \{0, 1, 3, 5\}$
$T_w$	Grupo de días del año de tipo $w$ , $w \in F$
$K$	Conjunto de diferentes tipos de días para el grupo $T_1$ , donde $K = \{WED, WEN, HOLD, HOLN\}$
$T_1^k$	Grupo de días del año de tipo $w = 1$ y tipo $k$ , con $k \in K$
$TA_{r,j}$	<i>Turnos Anuales</i> , representa el número de turnos teóricos de $S_j$ que cada médico de grupo $G_r$ debe de realizar al año, $r = 1, \dots, P$ ; $j = 1, \dots, L$
$TDE_{r,k}$	<i>Turnos Días Especiales</i> , representa el número de turnos teóricos que cada médico de grupo $G_r$ debe de realizar al año en los días $t \in T_1^k$ , $r = 1, \dots, P$ ; $k \in K$
$ATD_{w,j}$	<i>Asignación Turnos Diarios</i> , representa el número de turnos de $S_j$ que se asignan en cada día $t \in T_w$ , $w \in F$ ; $j = 1, \dots, L$

Parámetros auxiliares	
$I_{G_r}$	Intervalo de índices $i$ correspondientes a los médicos $M_i$ que pertenecen al grupo $G_r$ , $r = 1, \dots, P$ . $I_{G_r} = [A(r-1) + 1, A(r-1) + n_r],$ donde $A(r-1) = \sum_{\gamma=1}^{r-1} n_\gamma$
$L_r$	Conjunto de tipos de turnos $j$ que un médico de grupo $G_r$ no trabaja, $r = 1, \dots, P$ . $L_r = \begin{cases} 3, & r = 1, \\ 4, & r = 6, 7, \\ 0, & \text{en caso contrario} \end{cases}$
$SAB$	Grupo de días del año que son Sábado, excepto el día $t = 361$ . $SAB = \{t \in \{1, \dots, T\} : (t-1) \bmod 7 + 2 = 5 \wedge t \neq 361\}$
Variables de decisión	
$X_{i,j,t}$	Variable de decisión binaria que determina si un médico $M_i$ trabaja un turno de $S_j$ el día $t$
$d_{i,j}^+$	Variable meta positiva que mide el exceso del número de turnos de $S_j$ que un médico $M_i$ realiza en el año
$d_{i,j}^-$	Variable meta negativa que mide la falta del número de turnos de $S_j$ que un médico $M_i$ realiza en el año
$c_{i,k}^+$	Variable meta positiva que mide el exceso del número de turnos que un médico $M_i$ realiza al año en los días $t \in T_1^k$
$c_{i,k}^-$	Variable meta negativa que mide la falta del número de turnos que un médico $M_i$ realiza al año en los días $t \in T_1^k$

## B.2. Datos

En las siguientes tablas se recogen los datos del problema de planificación de turnos hospitalarios en un SU.

Parámetros	Valores
$N$	43 médicos
$P$	7 tipos de médicos
$L$	4 tipos de turnos
$T$	366 días

Tabla B.4: Valores de los parámetros iniciales.

	$n_r$	$h_r$
$G_1$	6	$1766 + 14$
$G_2$	24	$1766 + 14$
$G_3$	3	$1766 + 14$
$G_4$	6	$1766 + 14$
$G_5$	2	$1766 + 14$
$G_6$	1	$1314 + 14$
$G_7$	1	$1314 + 14$

Tabla B.5: Número de médicos  $n_r$  y horas totales  $h_r$  que debe de realizar cada médico de grupo  $G_r$  al año, para todo  $r = 1, \dots, 7$ . A la jornada anual máxima se le permite un margen de desviación de 14 horas, que se corresponde a un turno de  $S_2$ .

	$S_1$	$S_2$	$S_3$	$S_4$
$G_1$	30.82	105.06	0.00	15.22
$G_2$	15.34	51.64	45.55	15.22
$G_3$	7.18	23.48	69.55	15.22
$G_4$	11.26	37.56	57.55	15.22
$G_5$	13.29	44.63	51.55	15.22
$G_6$	12.20	40.48	33.89	0.00
$G_7$	8.12	26.45	45.89	0.00
$d_j$	8	14	19.2	3

Tabla B.6: Número de turnos teóricos de  $S_j$  que cada médico de grupo  $G_r$  debe de realizar al año. Cada elemento se representa con la variable  $TA_{r,j}$ , para todo  $r = 1, \dots, 7$ ;  $j = 1, \dots, 4$ . Y en la última fila, número de horas  $d_j$  que dura cada turno de  $S_j$ , para todo  $j = 1, \dots, 4$ .

	$T_1^{WED}$	$T_1^{WEN}$	$T_1^{HOLD}$	$T_1^{HOLN}$
$G_1$	24.48	0.00	3.77	0.00
$G_2$	11.53	12.94	1.77	1.99
$G_3$	4.71	19.76	0.73	3.04
$G_4$	8.12	16.35	1.25	2.52
$G_5$	9.83	14.65	1.51	2.25
$G_6$	8.58	9.63	1.32	1.48
$G_7$	5.17	13.04	0.80	2.01

Tabla B.7: Número de turnos teóricos que cada médico de grupo  $G_r$  debe de realizar al año en los días  $t \in T_1^k$ . Cada elemento se representa con la variable  $TDE_{r,k}$ , para todo  $r = 1, \dots, 7$ ;  $k \in K$ .

	$S_1$	$S_2$	$S_3$	$S_4$
$T_0$	3	7	5	0
$T_1$	0	5	5	0
$T_3$	3	7	5	26
$T_5$	2	7	5	0

Tabla B.8: Número de turnos de  $S_j$  que se asignan en cada día  $t \in T_w$ . Cada elemento se representa con la variable  $ATD_{w,j}$ , para todo  $w \in F$ ;  $j = 1, \dots, 4$ .

### B.3. Formulación matemática

El problema de planificación de turnos hospitalarios en un SU se modela como un problema de PLE binario mixto mediante el uso de las variables de decisión binarias  $X_{i,j,t}$  definidas en el Capítulo 4 como

$$X_{i,j,t} = \begin{cases} 1, & \text{si un médico } M_i \text{ trabaja un turno de } S_j \text{ el día } t, \\ 0, & \text{en caso contrario.} \end{cases} \quad \forall i = 1, \dots, N; j = 1, \dots, L; t = 1, \dots, T.$$

Las restricciones del modelo se dividen en fuertes y suaves. Las restricciones fuertes vienen definidas por las características de los datos del SU y por las medidas adoptadas por Ley. De esta manera, dentro de las restricciones fuertes se encuentran restricciones de: (a) disponibilidad horaria, (b) cubrimiento y (c) ergonomía. Por otra parte, se encuentran las restricciones suaves, las cuales se podrán incumplir sujetas unas penalizaciones en la función objetivo.

Para modelar las restricciones, definimos el intervalo  $I_{G_r}$  que contiene a los índices  $i$  correspondientes a los médicos  $M_i$  que pertenecen al grupo  $G_r$ , dado por

$$I_{G_r} = [A(r-1) + 1, A(r-1) + n_r],$$

donde  $A(r-1) = \sum_{\gamma=1}^{r-1} n_\gamma$ ,  $\forall r = 1, \dots, P$ . En particular, se asume que  $A(0) = \sum_{\gamma=1}^0 n_\gamma = 0$ .

#### Restricciones fuertes

Las restricciones fuertes empleadas en el problema son las siguientes:

- **Disponibilidad horaria.** Todo médico perteneciente al grupo  $G_r$  no puede trabajar más de  $h_r$  horas al año.

$$\sum_{t=1}^T \sum_{j=1}^L d_j X_{i,j,t} \leq h_r, \quad \forall r = 1, \dots, P; i \in I_{G_r}. \quad (1)$$

- **Cubrimiento.** Hace referencia a todas aquellas restricciones que modelan la forma en la que se asignan los turnos debido a las características del SU.

- (i) Ningún médico puede realizar más de un turno diario.

$$\sum_{j=1}^L X_{i,j,t} \leq 1, \quad \forall i = 1, \dots, N; t = 1, \dots, T. \quad (2)$$

- (ii) Asignación de los turnos en función del tipo de día.

$$\sum_{i=1}^N X_{i,j,t} = ATD_{w,j}, \quad \forall w \in F; t \in T_w; j = 1, \dots, L. \quad (3)$$

- (iii) Los médicos de grupo  $G_1$  no realizan turnos de  $S_3$  y, los médicos de los grupos  $G_6$ ,  $G_7$  no realizan turnos de  $S_4$ . Estas características se representan con ceros en la Tabla B.6. Para modelar estas restricciones definimos la función  $L_r$ , dada por

$$L_r = \begin{cases} 3, & r = 1, \\ 4, & r = 6, 7, \\ 0, & \text{en caso contrario,} \end{cases} \quad \forall r = 1, \dots, P.$$

que representa los tipos de turno  $j$  que un médico de grupo  $G_r$  no trabaja. El cero representa que se realizan todos los tipos de turnos. De esta forma, la restricción se modela como

$$\sum_{t=1}^T X_{i,j,t} = TA_{r,j}, \quad \forall r = 1, \dots, P; j \in L_r : j \neq 0; i \in I_{G_r}. \quad (4)$$

- (iv) Los médicos de grupo  $G_1$  no trabajan los fines de semana por la noche ni los días festivos por la noche. Estas particularidades se representan con ceros en la Tabla B.7. Tenemos que en los fines de semana por la noche y en los días festivos por la noche únicamente se trabajan turnos de  $S_3$ , ya que es el único de turno que tiene lugar por la noche. Además, por la restricción (4), se sigue que estos médicos no realizan turnos de  $S_3$ . Así pues, en particular, (4) contempla esta restricción.
- (v) Los médicos de los grupos  $G_6$ ,  $G_7$  no trabajan los días de tipo Miércoles con reunión, que son días de tipo  $w = 3$ .

$$\sum_{j=1}^{L-1} X_{i,j,t} \leq 0, \quad \forall r = 6, 7; t = 1, \dots, T : t \in T_3; i \in I_{G_r}. \quad (5)$$

Es suficiente que el índice  $j$  recorra únicamente el conjunto  $\{1, \dots, L-1\}$  ya que hemos impuesto en (4) que los médicos de estos grupos no realizan turnos de  $S_L$ , con  $L = 4$ .

- **Ergonomía.** Nos referimos a todas aquellas restricciones que modelan el número de días de descanso que se deben de seguir por Ley dependiendo el tipo de turno realizado. Un día de descanso significa que en ese día no se pueden realizar turnos de  $S_1$ ,  $S_2$ ,  $S_3$ . De esta forma, los días de descanso asignados a cada médico se deben de respetar, excepto cuando uno de los días de descanso coincida con un día de tipo Miércoles con reunión, donde se le asignará un turno de  $S_4$ . Por ejemplo, si un médico realiza un turno de  $S_2$ , entonces descansa el día siguiente. Sin embargo, si ese día de descanso coincide con un día de  $T_3$ , entonces se le asigna un turno de  $S_4$  y pierde el día de descanso. En caso contrario, tiene el día libre. Por tanto, es suficiente que el índice de los tipos de turno  $j$  recorra el conjunto  $\{1, \dots, L-1\}$ .

- (i) Cualquier médico que realice un turno de  $S_2$  descansa el día siguiente.

$$\sum_{j=1}^{L-1} X_{i,j,(t+1)} \leq 1 - X_{i,2,t}, \quad \forall i = 1, \dots, N; t = 1, \dots, T-1. \quad (6)$$

- (ii) Cualquier médico que realice un turno de  $S_3$  descansa los dos días siguientes. Tenemos las siguientes dos restricciones:

$$\begin{aligned} \sum_{j=1}^{L-1} [X_{i,j,(t+1)} + X_{i,j,(t+2)}] &\leq 2(1 - X_{i,3,t}), \quad \forall i = 1, \dots, N; t = 1, \dots, T-2, \\ \sum_{j=1}^{L-1} X_{i,j,(t+1)} &\leq 1 - X_{i,3,t}, \quad \forall i = 1, \dots, N; t = T-1. \end{aligned} \quad (*)$$

Estas dos restricciones se pueden expresar en una única donde el índice  $t$  recorre todo el conjunto  $\{1, \dots, T-1\}$ . Para ello, hacemos uso de la función característica definida por

$$\chi_A(x) = \begin{cases} 1, & \text{si } x \in A, \\ 0, & \text{en caso contrario,} \end{cases}$$

donde  $A$  es un conjunto cualquiera. De esta forma, para todo  $i = 1, \dots, N$ ;  $t = 1, \dots, T-1$ , las dos restricciones (\*) se pueden expresar como

$$\Upsilon(i, t) \leq v(i, t),$$

donde

$$\begin{aligned} \Upsilon(i, t) &= \sum_{j=1}^{L-1} [X_{i,j,(t+1)} + X_{i,j,(t+2)}] \chi_{\{1, \dots, T-2\}}(t) + \sum_{j=1}^{L-1} X_{i,j,(t+1)} \chi_{\{T-1\}}(t), \\ v(i, t) &= 2 (1 - X_{i,3,t}) \chi_{\{1, \dots, T-2\}}(t) + (1 - X_{i,3,t}) \chi_{\{T-1\}}(t). \end{aligned}$$

Desarrollando  $\Upsilon(i, t)$  tenemos que, para todo  $i = 1, \dots, N$ ;  $t = 1, \dots, T-1$

$$\begin{aligned} \Upsilon(i, t) &= \sum_{j=1}^{L-1} X_{i,j,(t+1)} \chi_{\{1, \dots, T-2\}}(t) + \sum_{j=1}^{L-1} X_{i,j,(t+2)} \chi_{\{1, \dots, T-2\}}(t) + \sum_{j=1}^{L-1} X_{i,j,(t+1)} \chi_{\{T-1\}}(t) \\ &= \sum_{j=1}^{L-1} X_{i,j,(t+1)} + \sum_{j=1}^{L-1} X_{i,j,(t+2)} \chi_{\{1, \dots, T-2\}}(t) \\ &= \sum_{j=1}^{L-1} [X_{i,j,(t+1)} + X_{i,j,(t+2)} \chi_{\{1, \dots, T-2\}}(t)]. \end{aligned}$$

Por otra parte, desarrollando  $v(i, t)$  se sigue que, para todo  $i = 1, \dots, N$ ;  $t = 1, \dots, T-1$

$$v(i, t) = 2 [\chi_{\{1, \dots, T-2\}}(t) + \frac{1}{2} \chi_{\{T-1\}}(t)] [1 - X_{i,3,t}].$$

Además, como

$$\chi_{\{1, \dots, T-2\}}(t) + \frac{1}{2} \chi_{\{T-1\}}(t) = \begin{cases} 1, & \text{si } t \in \{1, \dots, T-2\}, \\ \frac{1}{2}, & \text{si } t = T-1, \end{cases} \quad (**)$$

utilizando la definición alternativa de la función de *Heaviside* dada por

$$H(x) = \begin{cases} 0, & \text{si } x < 0, \\ \frac{1}{2}, & \text{si } x = 0, \\ 1, & \text{si } x > 0, \end{cases}$$

que permite una representación a través de la *función signo* como  $H(x) = \frac{1}{2}(1 + \text{sgn}(x))$ , se tiene entonces que

$$\chi_{\{1, \dots, T-2\}}(t) + \frac{1}{2} \chi_{\{T-1\}}(t) = H(T-1-t).$$

Además, para  $t > T-1$ , se verifica trivialmente la igualdad. Por tanto, el uso de la función de Heaviside permite extender la función (\*\*) para todo valor de  $t \geq 1$ . Así pues, para todo  $i = 1, \dots, N$ ;  $t = 1, \dots, T-1$  tenemos que

$$v(i, t) = 2 H(T-1-t) [1 - X_{i,3,t}].$$

Finalmente, las dos desigualdades iniciales (\*) se expresan de la siguiente manera,

$$\sum_{j=1}^{L-1} \left[ X_{i,j,(t+1)} + X_{i,j,(t+2)} \chi_{\{1,\dots,T-2\}}(t) \right] \leq 2 H(T-1-t) [1 - X_{i,3,t}], \quad (7)$$

$$\forall i = 1, \dots, N; t = 1, \dots, T-1.$$

- (iii) Ningún médico pueden trabajar dos fines de semanas consecutivos. Definimos el conjunto  $SAB$  de días del año que se corresponden a Sábado y, eliminamos el último Sábado del año 2020 que corresponde al día  $t = 361$ . El conjunto viene dado por

$$SAB = \{t \in \{1, \dots, T\} : (t-1) \bmod 7 + 2 = 5 \wedge t \neq 361\}.$$

Por otra parte, los fines de semana pertenecen al grupo  $T_1$  y, como se muestra en la Tabla B.8, para este grupo únicamente se pueden asignar turnos de  $S_2, S_3$ . Por tanto, tenemos la siguiente restricción:

$$\sum_{j \in \{1, \dots, L\} \setminus \{1,4\}} \left[ X_{i,j,(t+7)} + X_{i,j,(t+8)} \right] \leq 1 - \sum_{j \in \{1, \dots, L\} \setminus \{1,4\}} \left[ X_{i,j,t} + X_{i,j,(t+1)} \right], \quad (8)$$

$$\forall i = 1, \dots, N; t \in SAB.$$

### Restricciones suaves

Una vez que nos hemos asegurado del cumplimiento de las restricciones fuertes, con el uso de las variables metas  $d_{i,j}^+, d_{i,j}^-, c_{i,k}^+, c_{i,k}^-$  que toman valores en  $\mathbb{R}^+$  modelamos las restricciones suaves para que el número de turnos asignados entre los médicos sean lo más próximos posibles a los datos teóricos  $TA_{r,j}$  y  $TDE_{r,k}$  y, por tanto, la planificación sea equilibrada. Tenemos las siguientes restricciones:

- (i) Todos los médicos de un mismo grupo deben de realizar el mismo número de turnos de cada tipo al año.

$$\sum_{t=1}^T X_{i,j,t} + d_{i,j}^- - d_{i,j}^+ = TA_{r,j}, \quad \forall r = 1, \dots, P; i \in I_{G_r}; j = 1, \dots, L, \quad (9)$$

donde  $d_{i,j}^+$  y  $d_{i,j}^-$  son las variables metas que miden el exceso y la falta en el número de turnos de  $S_j$  que un médico  $M_i$  realiza en el año, respectivamente.

- (ii) Todos los médicos de un mismo grupo deben de realizar el mismo número de turnos al año en los días de tipo  $k$ , con  $k \in K$ .

$$\sum_{t \in T_1^k} \sum_{j=1}^L X_{i,j,t} + c_{i,k}^- - c_{i,k}^+ = TDE_{r,k}, \quad \forall r = 1, \dots, P; i \in I_{G_r}; k \in K, \quad (10)$$

donde  $c_{i,k}^+$  y  $c_{i,k}^-$  son las variables metas que miden el exceso y la falta en el número de turnos que un médico  $M_i$  realiza al año para los días  $t \in T_1^k$ , respectivamente.

La función objetivo se define para minimizar la suma de los excesos y las faltas en la asignación de los turnos y así, el reparto de turnos sea lo más equitativo posible. Con esta finalidad, definimos la función objetivo como el mínimo de la suma de todas las variables metas, es decir,

$$\text{Mín} \sum_{i=1}^N \left[ \sum_{j=1}^L (d_{i,j}^+ + d_{i,j}^-) + \sum_{k=1}^K (c_{i,k}^+ + c_{i,k}^-) \right]. \quad (11)$$

En la siguiente hoja se presenta la formulación completa del problema.

El problema de planificación de turnos hospitalarios en un SU se modeliza como un problema de PLE binario mixto y, consiste en minimizar la función objetivo (11) sujeto a las restricciones (1)-(10).

$$\text{Mín} \quad \sum_{i=1}^N \left[ \sum_{j=1}^L (d_{i,j}^+ + d_{i,j}^-) + \sum_{k=1}^K (c_{i,k}^+ + c_{i,k}^-) \right]$$

sujeto a:

$$\sum_{t=1}^T X_{i,j,t} + d_{i,j}^- - d_{i,j}^+ = TA_{r,j}, \quad \forall r = 1, \dots, P; i \in I_{G_r}; j = 1, \dots, L$$

$$\sum_{t \in T_1^k} \sum_{j=1}^L X_{i,j,t} + c_{i,k}^- - c_{i,k}^+ = TDE_{r,k}, \quad \forall r = 1, \dots, P; i \in I_{G_r}; k \in K$$

$$\sum_{t=1}^T \sum_{j=1}^L d_j X_{i,j,t} \leq h_r, \quad \forall r = 1, \dots, P; i \in I_{G_r}$$

$$\sum_{j=1}^L X_{i,j,t} \leq 1, \quad \forall i = 1, \dots, N; t = 1, \dots, T$$

$$\sum_{i=1}^N X_{i,j,t} = ATD_{w,j}, \quad \forall w \in F; t \in T_w; j = 1, \dots, L$$

$$\sum_{t=1}^T X_{i,j,t} = TA_{r,j}, \quad \forall r = 1, \dots, P; j \in L_r : j \neq 0; i \in I_{G_r}$$

$$\sum_{j=1}^{L-1} X_{i,j,t} \leq 0, \quad \forall r = 6, 7; t = 1, \dots, T : t \in T_3; i \in I_{G_r}$$

$$\sum_{j=1}^{L-1} X_{i,j,(t+1)} \leq 1 - X_{i,2,t}, \quad \forall i = 1, \dots, N; t = 1, \dots, T-1$$

$$\sum_{j=1}^{L-1} \left[ X_{i,j,(t+1)} + X_{i,j,(t+2)} \chi_{\{1, \dots, T-2\}}(t) \right] \leq 2H(T-1-t)[1 - X_{i,3,t}], \quad \forall i = 1, \dots, N; t = 1, \dots, T-1$$

$$\sum_{j \in \{1, \dots, L\} \setminus \{1,4\}} \left[ X_{i,j,(t+7)} + X_{i,j,(t+8)} \right] \leq 1 - \sum_{j \in \{1, \dots, L\} \setminus \{1,4\}} \left[ X_{i,j,t} + X_{i,j,(t+1)} \right], \quad \forall i = 1, \dots, N; t \in SAB$$

$$X_{i,j,t} \in \{0, 1\}, \quad \forall i = 1, \dots, N; j = 1, \dots, L; t = 1, \dots, T$$

$$d_{i,j}^+, d_{i,j}^-, c_{i,k}^+, c_{i,k}^- \in \mathbb{R}^+, \quad \forall i = 1, \dots, N; j = 1, \dots, L; k \in K$$



## B.4. Código de resolución en CPLEX

El código consta de tres archivos: (1) *AsignacionMedicos.dat*, donde se definen los datos del problema, (2) *AsignacionMedicos.mod*, donde se modela el problema de PLE binario mixto y (3) *AsignacionMedicosExcel.dat*, donde la solución obtenida se exporta a un libro de Excel previamente maquettato que hemos llamado *AsignacionMedicosSol.xlsx*.

### Código 1: AsignacionMedicos.dat

```

1 //Datos  $TA_{r,j}$ .
2 TurnosAnuales = [
3 [30.82 105.06 0.00 15.22]
4 [15.34 51.64 45.55 15.22]
5 [ 7.18 23.48 69.55 15.22]
6 [11.26 37.56 57.55 15.22]
7 [13.29 44.63 51.55 15.22]
8 [12.20 40.48 33.89 0.00]
9 [ 8.12 26.45 45.89 0.00]
10 ];
11
12 //Datos  $TDE_{r,k}$ .
13 TurnosDiasEspeciales = [
14 [24.48 0.00 3.77 0.00]
15 [11.53 12.94 1.77 1.99]
16 [ 4.71 19.76 0.73 3.04]
17 [ 8.12 16.35 1.25 2.52]
18 [ 9.83 14.65 1.51 2.25]
19 [ 8.58 9.63 1.32 1.48]
20 [ 5.17 13.04 0.80 2.01]
21 ];
22
23 //Datos  $ATD_{w,j}$ .
24 AsignacionTurnosDiarios = [
25 [3 7 5 0 ]
26 [0 5 5 0 ]
27 [3 7 5 26]
28 [2 7 5 0 ]
29 ];
30
31 //Calendario.
32 //Vector de 366 elementos que contiene los tipos de día  $w$  que corresponden a cada día del año.
33 Calendario =
34 [1, 5, 0, 1, 1, 1, 0, 0, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1,
    0, 0, 0, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5,
    0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0,
    0, 0, 1, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 0, 1, 1,
    1, 1, 1, 0, 0, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5, 1, 1, 1, 0, 0,
    0, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1,
    1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 0,
    5, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1,
    0, 0, 0, 5, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 0, 0, 5,
    0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0,
    0, 0, 5, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 0, 3, 5, 0,
    1, 1, 0, 0, 0, 5, 0, 1, 1, 1, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0,
    0, 5, 0, 1, 1, 0, 0, 0, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1, 1, 0, 0, 3, 5, 0, 1,
    1, 0, 0, 3, 5, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 5, 0, 1, 1, 0, 0, 3,
    5, 0, 1, 1, 0, 0, 0, 5, 1, 1, 1, 0, 0, 0, 5];

```

## Código 2: AsignacionMedicos.mod

```

1  //////////////////////////////////1.- DECLARACIÓN DE LOS DATOS////////////////////////////////////
2
3
4  int NumeroDiasTotal = 366;    //Número de días en el año 2020.
5  int NumeroMedicosTotal = 43; //Número total de médicos.
6
7  range RangoMedicos = 1..NumeroMedicosTotal; //Rango de índices de los médicos.
8  range RangoDias = 1..NumeroDiasTotal;      //Rango de días del año.
9
10 range RangoGrupo1 = 1..6;    //Rango de índices correspondiente a los médicos de G1.
11 range RangoGrupo2 = 7..30;   //Rango de índices correspondiente a los médicos de G2.
12 range RangoGrupo3 = 31..33;  //Rango de índices correspondiente a los médicos de G3.
13 range RangoGrupo4 = 34..39;  //Rango de índices correspondiente a los médicos de G4.
14 range RangoGrupo5 = 40..41;  //Rango de índices correspondiente a los médicos de G5.
15
16 //Grupo de médicos.
17 {string} GrupoMedicos = {"G1", "G2", "G3", "G4", "G5", "G6", "G7"};
18
19 //Grupo de turnos.
20 {string} GrupoTiposTurno = {"S1", "S2", "S3", "S4"};
21
22 //Conjunto K.
23 {string} ConjuntoDiasEspeciales = {"WED", "WEN", "HOLD", "HOLN"};
24
25 //Conjunto F.
26 {int} ConjuntoTiposDias = {0, 1, 3, 5};
27
28
29 //Declaramos la matriz TurnosAnuales que contiene los datos  $TA_{r,j}$  definidos en el archivo .dat.
30 float TurnosAnuales[GrupoMedicos][GrupoTiposTurno] = ...;
31
32 //Declaramos la matriz TurnosDiasEspeciales que contiene los datos  $TDE_{r,k}$  definidos en el
33 //archivo .dat.
34 float TurnosDiasEspeciales[GrupoMedicos][ConjuntoDiasEspeciales] = ...;
35
36 //Declaramos el vector Calendario definido en el archivo .dat.
37 int Calendario[RangoDias] = ...;
38
39 //Declaramos la matriz AsignacionTurnosDiarios que contiene los datos  $ATD_{w,j}$  definidos en el
40 //archivo .dat.
41 int AsignacionTurnosDiarios[1..4][GrupoTiposTurno] = ...;
42
43 //Definimos los conjuntos auxiliares ConjuntoDiasTipo1 y ConjuntoDiasTipo3 que contienen todos
44 //los días del año que son de tipo 1 y 3, respectivamente.
45 {int} ConjuntoDiasTipo1 = { dia | dia in RangoDias : Calendario[dia] == 1};
46 {int} ConjuntoDiasTipo3 = { dia | dia in RangoDias : Calendario[dia] == 3};
47
48
49 //Definimos el conjunto auxiliar ConjuntoDiasFestivos que contiene todos los días festivos del
50 //año. Los días festivos serán aquellos días del conjunto ConjuntoDiasTipo1 tales que no
51 //coincidan en Sábado ni en Domingo.
52 //Trabajamos en modulo 7, pero el índice del vector Calendario va de 1..366. Restamos 1 para
53 //que vaya de 0..365. Hacemos (dia - 1) mod 7 para conocer el día de la semana que le
54 //corresponde, pero el año empieza en Miércoles, entonces sumamos 2 por ese desplazamiento.
55 {int} ConjuntoDiasFestivos = {dia | dia in ConjuntoDiasTipo1 : ((dia - 1) mod 7
    + 2) not in {5,6}};
56
57 //Definimos el conjunto auxiliar ConjuntoDiasFindeSemana que contiene todos los días del año
58 //correspondientes a fines de semana. Estos días serán los que hay en el conjunto
59 //ConjuntoDiasTipo1 menos los del conjunto ConjuntoDiasFestivos.
60 {int} ConjuntoDiasFindeSemana = ConjuntoDiasTipo1 diff ConjuntoDiasFestivos;
61

```

```

62 //Definimos el rango auxiliar RangoFindeSemana que contiene el número de fines de semana en el
63 //año. Restamos 1 a ftoi(card(ConjuntoDiasFindeSemana)/2) porque queremos empezar el rango en 0
64 //en vez de en 1. Restamos otro 1 para quitar el caso del último fin de semana del año.
65 range RangoFindeSemana = 0..ftoi(card(ConjuntoDiasFindeSemana)/2) - 2;
66
67
68
69 //////////////////////////////////////////////////////////////////2.- DEFINICIÓN DE LAS VARIABLES DE DECISIÓN////////////////////////////////////
70
71
72 //2.1.- VARIABLES DE DECISIÓN-----
73
74 //Definimos la matrices de variables booleanas que se corresponden a las
75 //variables de decisión  $X_{i,1,t}$ ,  $X_{i,2,t}$ ,  $X_{i,3,t}$ ,  $X_{i,4,t}$ , respectivamente.
76 dvar boolean AsignacionTurnosS1[RangoMedicos][RangoDias];
77 dvar boolean AsignacionTurnosS2[RangoMedicos][RangoDias];
78 dvar boolean AsignacionTurnosS3[RangoMedicos][RangoDias];
79 dvar boolean AsignacionTurnosS4[RangoMedicos][RangoDias];
80
81 //Definimos la matrices de variables reales no negativas correspondientes a las variables metas
82 // $d_{i,j}^-$ ,  $d_{i,j}^+$ , para  $j=1,\dots,4$ . En la primera columna de cada matriz se almacenan las variables
83 //metas negativas y, en la segunda las variables metas positivas.
84 dvar float+ MetasTurnosS1[RangoMedicos][1..2];
85 dvar float+ MetasTurnosS2[RangoMedicos][1..2];
86 dvar float+ MetasTurnosS3[RangoMedicos][1..2];
87 dvar float+ MetasTurnosS4[RangoMedicos][1..2];
88
89 //Definimos la matrices de variables reales no negativas correspondientes a las variables metas
90 // $c_{i,k}^-$ ,  $c_{i,k}^+$ , para  $k \in K$ . Al igual que antes, en la primera columna se almacenan las variables
91 //metas negativas y, en la segunda las positivas.
92 dvar float+ MetasWED[RangoMedicos][1..2];
93 dvar float+ MetasWEN[RangoMedicos][1..2];
94 dvar float+ MetasHOLD[RangoMedicos][1..2];
95 dvar float+ MetasHOLN[RangoMedicos][1..2];
96
97
98 //2.2.- EXPRESIONES DE LAS VARIABLES DE DECISIÓN-----
99 //Definimos las siguientes expresiones que se emplearán en las restricciones del punto 4.2.
100
101 //Sumamos para cada médico de  $G_r$  los turnos de  $S_1$  que ha realizado en el año.
102 dexpr float G1S1[i in RangoGrupo1] = sum(j in RangoDias) AsignacionTurnosS1[i][j];
103 dexpr float G2S1[i in RangoGrupo2] = sum(j in RangoDias) AsignacionTurnosS1[i][j];
104 dexpr float G3S1[i in RangoGrupo3] = sum(j in RangoDias) AsignacionTurnosS1[i][j];
105 dexpr float G4S1[i in RangoGrupo4] = sum(j in RangoDias) AsignacionTurnosS1[i][j];
106 dexpr float G5S1[i in RangoGrupo5] = sum(j in RangoDias) AsignacionTurnosS1[i][j];
107 dexpr float G6S1 = sum(j in RangoDias) AsignacionTurnosS1[42][j];
108 dexpr float G7S1 = sum(j in RangoDias) AsignacionTurnosS1[43][j];
109
110 //Sumamos para cada médico de  $G_r$  los turnos de  $S_2$  que ha realizado en el año.
111 dexpr float G1S2[i in RangoGrupo1] = sum(j in RangoDias) AsignacionTurnosS2[i][j];
112 dexpr float G2S2[i in RangoGrupo2] = sum(j in RangoDias) AsignacionTurnosS2[i][j];
113 dexpr float G3S2[i in RangoGrupo3] = sum(j in RangoDias) AsignacionTurnosS2[i][j];
114 dexpr float G4S2[i in RangoGrupo4] = sum(j in RangoDias) AsignacionTurnosS2[i][j];
115 dexpr float G5S2[i in RangoGrupo5] = sum(j in RangoDias) AsignacionTurnosS2[i][j];

```

```

    j];
116 dexpr float G6S2 = sum(j in RangoDias) AsignacionTurnosS2[42][j];
117 dexpr float G7S2 = sum(j in RangoDias) AsignacionTurnosS2[43][j];
118
119 //Sumamos para cada médico de  $G_r$  los turnos de  $S_3$  que ha realizado en el año.
120 dexpr float G1S3[i in RangoGrupo1] = sum(j in RangoDias) AsignacionTurnosS3[i][
    j];
121 dexpr float G2S3[i in RangoGrupo2] = sum(j in RangoDias) AsignacionTurnosS3[i][
    j];
122 dexpr float G3S3[i in RangoGrupo3] = sum(j in RangoDias) AsignacionTurnosS3[i][
    j];
123 dexpr float G4S3[i in RangoGrupo4] = sum(j in RangoDias) AsignacionTurnosS3[i][
    j];
124 dexpr float G5S3[i in RangoGrupo5] = sum(j in RangoDias) AsignacionTurnosS3[i][
    j];
125 dexpr float G6S3 = sum(j in RangoDias) AsignacionTurnosS3[42][j];
126 dexpr float G7S3 = sum(j in RangoDias) AsignacionTurnosS3[43][j];
127
128 //Sumamos para cada médico de  $G_r$  los turnos de  $S_4$  que ha realizado en el año.
129 dexpr float G1S4[i in RangoGrupo1] = sum(j in RangoDias) AsignacionTurnosS4[i][
    j];
130 dexpr float G2S4[i in RangoGrupo2] = sum(j in RangoDias) AsignacionTurnosS4[i][
    j];
131 dexpr float G3S4[i in RangoGrupo3] = sum(j in RangoDias) AsignacionTurnosS4[i][
    j];
132 dexpr float G4S4[i in RangoGrupo4] = sum(j in RangoDias) AsignacionTurnosS4[i][
    j];
133 dexpr float G5S4[i in RangoGrupo5] = sum(j in RangoDias) AsignacionTurnosS4[i][
    j];
134 dexpr float G6S4 = sum(j in RangoDias) AsignacionTurnosS4[42][j];
135 dexpr float G7S4 = sum(j in RangoDias) AsignacionTurnosS4[43][j];
136
137
138 //Sumamos para cada médico de  $G_r$  los turnos que ha realizado en los días de tipo WED.
139 //En los días de tipo WED únicamente se realizan turnos de  $S_2$ .
140 dexpr float G1WED[i in RangoGrupo1] = sum(j in ConjuntoDiasFindeSemana)
    AsignacionTurnosS2[i][j];
141 dexpr float G2WED[i in RangoGrupo2] = sum(j in ConjuntoDiasFindeSemana)
    AsignacionTurnosS2[i][j];
142 dexpr float G3WED[i in RangoGrupo3] = sum(j in ConjuntoDiasFindeSemana)
    AsignacionTurnosS2[i][j];
143 dexpr float G4WED[i in RangoGrupo4] = sum(j in ConjuntoDiasFindeSemana)
    AsignacionTurnosS2[i][j];
144 dexpr float G5WED[i in RangoGrupo5] = sum(j in ConjuntoDiasFindeSemana)
    AsignacionTurnosS2[i][j];
145 dexpr float G6WED = sum(j in ConjuntoDiasFindeSemana) AsignacionTurnosS2[42][j
    ];
146 dexpr float G7WED = sum(j in ConjuntoDiasFindeSemana) AsignacionTurnosS2[43][j
    ];
147
148 //Sumamos para cada médico de  $G_r$  los turnos que ha realizado en los días de tipo WEN.
149 //En los días de tipo WEN únicamente se realizan turnos de  $S_3$ .
150 //Los médicos de  $G_1$  no hacen turnos de  $S_3$ , no los consideramos.
151 dexpr float G2WEN[i in RangoGrupo2] = sum(j in ConjuntoDiasFindeSemana)
    AsignacionTurnosS3[i][j];
152 dexpr float G3WEN[i in RangoGrupo3] = sum(j in ConjuntoDiasFindeSemana)
    AsignacionTurnosS3[i][j];
153 dexpr float G4WEN[i in RangoGrupo4] = sum(j in ConjuntoDiasFindeSemana)
    AsignacionTurnosS3[i][j];
154 dexpr float G5WEN[i in RangoGrupo5] = sum(j in ConjuntoDiasFindeSemana)
    AsignacionTurnosS3[i][j];
155 dexpr float G6WEN = sum(j in ConjuntoDiasFindeSemana) AsignacionTurnosS3[42][j
    ];

```

```

156 dexpr float G7WEN = sum(j in ConjuntoDiasFindeSemana) AsignacionTurnosS3[43][j
    ];
157
158 //Sumamos para cada médico de  $G_r$  los turnos que ha realizado en los días de tipo HOLD.
159 //En los días de tipo HOLD únicamente se realizan turnos de  $S_2$ .
160 dexpr float G1HOLD[i in RangoGrupo1] = sum(j in ConjuntoDiasFestivos)
    AsignacionTurnosS2[i][j];
161 dexpr float G2HOLD[i in RangoGrupo2] = sum(j in ConjuntoDiasFestivos)
    AsignacionTurnosS2[i][j];
162 dexpr float G3HOLD[i in RangoGrupo3] = sum(j in ConjuntoDiasFestivos)
    AsignacionTurnosS2[i][j];
163 dexpr float G4HOLD[i in RangoGrupo4] = sum(j in ConjuntoDiasFestivos)
    AsignacionTurnosS2[i][j];
164 dexpr float G5HOLD[i in RangoGrupo5] = sum(j in ConjuntoDiasFestivos)
    AsignacionTurnosS2[i][j];
165 dexpr float G6HOLD = sum(j in ConjuntoDiasFestivos) AsignacionTurnosS2[42][j];
166 dexpr float G7HOLD = sum(j in ConjuntoDiasFestivos) AsignacionTurnosS2[43][j];
167
168 //Sumamos para cada médico de  $G_r$  los turnos que ha realizado en los días de tipo HOLN.
169 //En los días de tipo HOLN únicamente se realizan turnos de  $S_3$ .
170 //Los médicos de  $G_1$  no hacen turnos de  $S_3$ , no los consideramos.
171 dexpr float G2HOLN[i in RangoGrupo2] = sum(j in ConjuntoDiasFestivos)
    AsignacionTurnosS3[i][j];
172 dexpr float G3HOLN[i in RangoGrupo3] = sum(j in ConjuntoDiasFestivos)
    AsignacionTurnosS3[i][j];
173 dexpr float G4HOLN[i in RangoGrupo4] = sum(j in ConjuntoDiasFestivos)
    AsignacionTurnosS3[i][j];
174 dexpr float G5HOLN[i in RangoGrupo5] = sum(j in ConjuntoDiasFestivos)
    AsignacionTurnosS3[i][j];
175 dexpr float G6HOLN = sum(j in ConjuntoDiasFestivos) AsignacionTurnosS3[42][j];
176 dexpr float G7HOLN = sum(j in ConjuntoDiasFestivos) AsignacionTurnosS3[43][j];
177
178
179
180 ///////////////////////////////////////////////////3.- FUNCIÓN OBJETIVO////////////////////////////////////
181 //En adelante, enumeramos las restricciones siguiendo los índices empleados en el Anexo B.3.
182
183 //RESTRICCIÓN (11):
184 minimize
185     sum(i in RangoMedicos) sum(j in 1..2) ( MetasTurnosS1[i][j] + MetasTurnosS2[i
        ][j] + MetasTurnosS3[i][j] + MetasTurnosS4[i][j] + MetasHOLN[i][j] +
        MetasHOLD[i][j] + MetasWEN[i][j] + MetasWED[i][j]);
186
187
188
189 ///////////////////////////////////////////////////4.- RESTRICCIONES////////////////////////////////////
190
191 subject to{
192
193 //4.1.- RESTRICCIONES FUERTES-----
194
195 //4.1.1.- RESTRICCIÓN (1): Disponibilidad horaria.
196
197 //Para los médicos de  $G_1, \dots, G_5$ :
198 forall(i in 1..41)
199     Restriccion_NumeroHorasTotal1:
200     sum(j in RangoDias) 8*AsignacionTurnosS1[i][j] + sum(j in RangoDias) 14*
        AsignacionTurnosS2[i][j] + sum(j in RangoDias) 19.2*AsignacionTurnosS3[i][j]
        + sum(j in RangoDias) 3*AsignacionTurnosS4[i][j] <= 1766 + 14;
201
202 //Para los médicos de  $G_6, G_7$ :
203 //No incluimos las horas del turno de  $S_4$  ya que no realizan este tipo de turno.
204 forall(i in 42..43)

```

```

205 Restriccion_NumeroHorasTotal2:
206 sum(j in RangoDias) 8*AsignacionTurnosS1[i][j] + sum(j in RangoDias) 14*
    AsignacionTurnosS2[i][j] + sum(j in RangoDias) 19.2*AsignacionTurnosS3[i][j]
    <= 1314 + 14;
207
208
209 //4.1.2.- RESTRICCIÓN (2): Ningún médico puede realizar más de un turno diario.
210
211 forall(j in RangoDias, i in RangoMedicos)
212     Restriccion_UnturnoMedico:
213     AsignacionTurnosS1[i][j] + AsignacionTurnosS2[i][j] + AsignacionTurnosS3[i][j]
        + AsignacionTurnosS4[i][j] <= 1;
214
215
216 //4.1.3.- RESTRICCIÓN (3): Asignación de los turnos en función del tipo de día w.
217
218 //Vamos a recorrer todos los días del año.
219 //A cada día le corresponde un tipo de día w que puede ser: 0, 1, 3, 5.
220 //En la matriz AsignacionTurnosDiarios tenemos los turnos que corresponden a cada tipo de día.
221 //Realizamos por separado las asignaciones para cada tipo de turno Sj.
222 //
223 //Por ejemplo:
224 //Para las asignaciones del número de turnos de S1,
225 //Fijado un día j, queremos movernos en la columna S1 de la matriz AsignacionTurnosDiarios y
226 //tomar el elemento correspondiente al tipo de día.
227 //Para ello hacemos:
228 //El tipo de día lo obtenemos con Calendario[j] y, con
229 //ord(ConjuntoTiposDias, Calendario[j]) + 1 convertimos el tipo de día a su correspondiente
230 //fila, de forma que conseguimos realizar la siguiente asignación,
231 //
232 // Tipo de día          Fila    S1 S2 ...
233 //      0      ----->    1    [ . . ... ]
234 //      1      ----->    2    [ . . ... ]
235 //      3      ----->    3    [ . . ... ]
236 //      5      ----->    4    [ . . ... ]
237
238 //Asignación de turnos de S1:
239 forall(j in RangoDias)
240     Restriccion_S1:
241     sum(i in RangoMedicos) AsignacionTurnosS1[i][j] == AsignacionTurnosDiarios[
        ord(ConjuntoTiposDias, Calendario[j])+1]["S1"];
242
243 //Asignación de turnos de S2:
244 forall(j in RangoDias)
245     Restriccion_S2:
246     sum(i in RangoMedicos) AsignacionTurnosS2[i][j] == AsignacionTurnosDiarios[
        ord(ConjuntoTiposDias, Calendario[j])+1]["S2"];
247
248 //Asignación de turnos de S3:
249 forall(j in RangoDias)
250     Restriccion_S3:
251     sum(i in RangoMedicos) AsignacionTurnosS3[i][j] == AsignacionTurnosDiarios[
        ord(ConjuntoTiposDias, Calendario[j])+1]["S3"];
252
253 //Asignación de turnos de S4:
254 forall(j in RangoDias)
255     Restriccion_S4:
256     sum(i in RangoMedicos) AsignacionTurnosS4[i][j] == AsignacionTurnosDiarios[
        ord(ConjuntoTiposDias, Calendario[j])+1]["S4"];
257
258
259 //4.1.4.- RESTRICCIÓN (5): Los médicos de los grupos G6, G7 no trabajan los días de tipo
260 //Miércoles con reunión.

```

```

261
262 //Recorremos los días almacenados en el conjunto ConjuntoDiasTipo3, que son todos los días
263 //correspondientes al tipo de día  $w=3$  e imponemos que no se les asignen turnos de  $S_1$ ,  $S_2$ ,  $S_3$ .
264 forall(i in 42..43, j in ConjuntoDiasTipo3)
265     Restriccion_G6_G7MiercolesReunion:
266     AsignacionTurnosS1[i][j] + AsignacionTurnosS2[i][j] + AsignacionTurnosS3[i][j]
267         ] <= 0;
268
269 //4.1.5.- RESTRICCIÓN (6): Cualquier médico que realice un turno de  $S_2$  descansa el día
270 //siguiente.
271
272 forall(i in RangoMedicos, j in 1..NumeroDiasTotal-1)
273     Restriccion_DescansoS2:
274     AsignacionTurnosS1[i][j+1] + AsignacionTurnosS2[i][j+1] + AsignacionTurnosS3[
275         i][j+1] <= 1 - AsignacionTurnosS2[i][j];
276
277 //4.1.6.- RESTRICCIONES (*): Cualquier médico que realice un turno de  $S_3$  descansa los dos días
278 //siguientes. Empleamos las restricciones de (*) que son equivalentes a la restricción (7).
279
280 forall(i in RangoMedicos, j in 1..NumeroDiasTotal-1)
281     if(j != NumeroDiasTotal-1)
282         Restriccion_DescansoS3:
283         sum(l in j+1..j+2)(AsignacionTurnosS1[i][l] + AsignacionTurnosS2[i][l] +
284             AsignacionTurnosS3[i][l]) <= 2 - 2*AsignacionTurnosS3[i][j];
285     else
286         Restriccion_DescansoS3_Dia366:
287         AsignacionTurnosS1[i][j+1] + AsignacionTurnosS2[i][j+1] +
288             AsignacionTurnosS3[i][j+1] <= 1 - AsignacionTurnosS3[i][j];
289
290 //4.1.7.- RESTRICCIÓN (8): Ningún médico pueden trabajar dos fines de semanas consecutivos.
291 //Utilizando el rango auxiliar RangoFindeSemana, tenemos que dado un  $j \in \text{RangoFindeSemana}$ :
292 //item(ConjuntoDiasFindeSemana, 2*j) obtenemos un Sábado,
293 //item(ConjuntoDiasFindeSemana, (2*j + 1)) obtenemos un Domingo,
294 //item(ConjuntoDiasFindeSemana, (2*j + 2)) obtenemos un Sábado de la próxima semana,
295 //item(ConjuntoDiasFindeSemana, (2*j + 3)) obtenemos un Domingo de la próxima semana.
296 //Esta formulación es equivalente a la mostrada en el Anexo B.3 empleando el conjunto SAB.
297 forall(i in RangoMedicos, j in RangoFindeSemana)
298     Restriccion_FinesDeSemanaConsecutivos:
299     AsignacionTurnosS2[i][item(ConjuntoDiasFindeSemana, (2*j + 2))] +
300         AsignacionTurnosS2[i][item(ConjuntoDiasFindeSemana, (2*j + 3))] +
301         AsignacionTurnosS3[i][item(ConjuntoDiasFindeSemana, (2*j + 2))] +
302         AsignacionTurnosS3[i][item(ConjuntoDiasFindeSemana, (2*j + 3))] <= 1 -
303         AsignacionTurnosS2[i][item(ConjuntoDiasFindeSemana, 2*j)] -
304         AsignacionTurnosS2[i][item(ConjuntoDiasFindeSemana, (2*j + 1))] -
305         AsignacionTurnosS3[i][item(ConjuntoDiasFindeSemana, 2*j)] -
306         AsignacionTurnosS3[i][item(ConjuntoDiasFindeSemana, (2*j + 1))];
307
308 //4.2.- RESTRICCIONES SUAVES-----
309
310 //4.2.1.- RESTRICCIÓN (9): Todos los médicos de un mismo grupo deben de realizar el mismo número
311 //de turnos de cada tipo al año. Imponemos las restricciones para cada grupo de médicos.
312
313 //Médicos de  $G_1$ :
314 forall(i in RangoGrupo1)
315     Restriccion_G1S1:
316     G1S1[i] + MetasTurnosS1[i][1] - MetasTurnosS1[i][2] == TurnosAnuales["G1"]["
317         S1"];

```

```

312 forall(i in RangoGrupo1)
313   Restriccion_G1S2:
314   G1S2[i] + MetasTurnosS2[i][1] - MetasTurnosS2[i][2] == TurnosAnuales["G1"]["
    S2"];
315 forall(i in RangoGrupo1)
316   Restriccion_G1S4:
317   G1S4[i] + MetasTurnosS4[i][1] - MetasTurnosS4[i][2] == TurnosAnuales["G1"]["
    S4"];
318
319 //RESTRICCIÓN (4): Imponemos que los médicos de  $G_1$  no hagan turnos de  $S_3$ .
320 forall(i in RangoGrupo1)
321   Restriccion_G1S3:
322   G1S3[i] == TurnosAnuales["G1"]["S3"];
323
324 //Médicos de  $G_2$ :
325 forall(i in RangoGrupo2)
326   Restriccion_G2S1:
327   G2S1[i] + MetasTurnosS1[i][1] - MetasTurnosS1[i][2] == TurnosAnuales["G2"]["
    S1"];
328 forall(i in RangoGrupo2)
329   Restriccion_G2S2:
330   G2S2[i] + MetasTurnosS2[i][1] - MetasTurnosS2[i][2] == TurnosAnuales["G2"]["
    S2"];
331 forall(i in RangoGrupo2)
332   Restriccion_G2S3:
333   G2S3[i] + MetasTurnosS3[i][1] - MetasTurnosS3[i][2] == TurnosAnuales["G2"]["
    S3"];
334 forall(i in RangoGrupo2)
335   Restriccion_G2S4:
336   G2S4[i] + MetasTurnosS4[i][1] - MetasTurnosS4[i][2] == TurnosAnuales["G2"]["
    S4"];
337
338 //Médicos de  $G_3$ :
339 forall(i in RangoGrupo3)
340   Restriccion_G3S1:
341   G3S1[i] + MetasTurnosS1[i][1] - MetasTurnosS1[i][2] == TurnosAnuales["G3"]["
    S1"];
342 forall(i in RangoGrupo3)
343   Restriccion_G3S2:
344   G3S2[i] + MetasTurnosS2[i][1] - MetasTurnosS2[i][2] == TurnosAnuales["G3"]["
    S2"];
345 forall(i in RangoGrupo3)
346   Restriccion_G3S3:
347   G3S3[i] + MetasTurnosS3[i][1] - MetasTurnosS3[i][2] == TurnosAnuales["G3"]["
    S3"];
348 forall(i in RangoGrupo3)
349   Restriccion_G3S4:
350   G3S4[i] + MetasTurnosS4[i][1] - MetasTurnosS4[i][2] == TurnosAnuales["G3"]["
    S4"];
351
352 //Médicos de  $G_4$ :
353 forall(i in RangoGrupo4)
354   Restriccion_G4S1:
355   G4S1[i] + MetasTurnosS1[i][1] - MetasTurnosS1[i][2] == TurnosAnuales["G4"]["
    S1"];
356 forall(i in RangoGrupo4)
357   Restriccion_G4S2:
358   G4S2[i] + MetasTurnosS2[i][1] - MetasTurnosS2[i][2] == TurnosAnuales["G4"]["
    S2"];
359 forall(i in RangoGrupo4)
360   Restriccion_G4S3:
361   G4S3[i] + MetasTurnosS3[i][1] - MetasTurnosS3[i][2] == TurnosAnuales["G4"]["
    S3"];

```



```

362 forall(i in RangoGrupo4)
363     Restriccion_G4S4:
364     G4S4[i] + MetasTurnosS4[i][1] - MetasTurnosS4[i][2] == TurnosAnuales["G4"]["
        S4"];
365
366 //Médicos de G5:
367 forall(i in RangoGrupo5)
368     Restriccion_G5S1:
369     G5S1[i] + MetasTurnosS1[i][1] - MetasTurnosS1[i][2] == TurnosAnuales["G5"]["
        S1"];
370 forall(i in RangoGrupo5)
371     Restriccion_G5S2:
372     G5S2[i] + MetasTurnosS2[i][1] - MetasTurnosS2[i][2] == TurnosAnuales["G5"]["
        S2"];
373 forall(i in RangoGrupo5)
374     Restriccion_G5S3:
375     G5S3[i] + MetasTurnosS3[i][1] - MetasTurnosS3[i][2] == TurnosAnuales["G5"]["
        S3"];
376 forall(i in RangoGrupo5)
377     Restriccion_G5S4:
378     G5S4[i] + MetasTurnosS4[i][1] - MetasTurnosS4[i][2] == TurnosAnuales["G5"]["
        S4"];
379
380 //Médicos de G6:
381 Restriccion_G6S1:
382 G6S1 + MetasTurnosS1[42][1] - MetasTurnosS1[42][2] == TurnosAnuales["G6"]["S1"
    ];
383 Restriccion_G6S2:
384 G6S2 + MetasTurnosS2[42][1] - MetasTurnosS2[42][2] == TurnosAnuales["G6"]["S2"
    ];
385 Restriccion_G6S3:
386 G6S3 + MetasTurnosS3[42][1] - MetasTurnosS3[42][2] == TurnosAnuales["G6"]["S3"
    ];
387
388 //RESTRICCIÓN (4): Imponemos que los médicos de G6 no hagan turnos de S4.
389 Restriccion_G6S4:
390 G6S4 == TurnosAnuales["G6"]["S4"];
391
392 //Médicos de G7:
393 Restriccion_G7S1:
394 G7S1 + MetasTurnosS1[43][1] - MetasTurnosS1[43][2] == TurnosAnuales["G7"]["S1"
    ];
395 Restriccion_G7S2:
396 G7S2 + MetasTurnosS2[43][1] - MetasTurnosS2[43][2] == TurnosAnuales["G7"]["S2"
    ];
397 Restriccion_G7S3:
398 G7S3 + MetasTurnosS3[43][1] - MetasTurnosS3[43][2] == TurnosAnuales["G7"]["S3"
    ];
399
400 //RESTRICCIÓN (4): Imponemos que los médicos de G7 no hagan turnos de S4.
401 Restriccion_G7S4:
402 G7S4 == TurnosAnuales["G7"]["S4"];
403
404
405 //4.2.2.- RESTRICCIÓN (10): Todos los médicos de un mismo grupo deben de realizar el mismo
406 //número de turnos al año en los días de tipo  $k$ , con  $k \in K$ . Imponemos las restricciones para
407 //cada grupo de médicos.
408
409 //Médicos de G1:
410 //No incluimos los tipos de días WEN, HOLN, ya que no realizan turnos esos días.
411 forall(i in RangoGrupo1)
412     Restriccion_G1HOLD:
413     G1HOLD[i] + MetasHOLD[i][1] - MetasHOLD[i][2] == TurnosDiasEspeciales["G1"]["

```

```

    HOLD"];
414 forall(i in RangoGrupo1)
415   Restriccion_G1WED:
416   G1WED[i] + MetasWED[i][1] - MetasWED[i][2] == TurnosDiasEspeciales["G1"]["WED
    "];
417
418 //Médicos de G2:
419 forall(i in RangoGrupo2)
420   Restriccion_G2HOLN:
421   G2HOLN[i] + MetasHOLN[i][1] - MetasHOLN[i][2] == TurnosDiasEspeciales["G2"]["
    HOLN"];
422 forall(i in RangoGrupo2)
423   Restriccion_G2HOLD:
424   G2HOLD[i] + MetasHOLD[i][1] - MetasHOLD[i][2] == TurnosDiasEspeciales["G2"]["
    HOLD"];
425 forall(i in RangoGrupo2)
426   Restriccion_G2WEN:
427   G2WEN[i] + MetasWEN[i][1] - MetasWEN[i][2] == TurnosDiasEspeciales["G2"]["WEN
    "];
428 forall(i in RangoGrupo2)
429   Restriccion_G2WED:
430   G2WED[i] + MetasWED[i][1] - MetasWED[i][2] == TurnosDiasEspeciales["G2"]["WED
    "];
431
432 //Médicos de G3:
433 forall(i in RangoGrupo3)
434   Restriccion_G3HOLN:
435   G3HOLN[i] + MetasHOLN[i][1] - MetasHOLN[i][2] == TurnosDiasEspeciales["G3"]["
    HOLN"];
436 forall(i in RangoGrupo3)
437   Restriccion_G3HOLD:
438   G3HOLD[i] + MetasHOLD[i][1] - MetasHOLD[i][2] == TurnosDiasEspeciales["G3"]["
    HOLD"];
439 forall(i in RangoGrupo3)
440   Restriccion_G3WEN:
441   G3WEN[i] + MetasWEN[i][1] - MetasWEN[i][2] == TurnosDiasEspeciales["G3"]["WEN
    "];
442 forall(i in RangoGrupo3)
443   Restriccion_G3WED:
444   G3WED[i] + MetasWED[i][1] - MetasWED[i][2] == TurnosDiasEspeciales["G3"]["WED
    "];
445
446 //Médicos de G4:
447 forall(i in RangoGrupo4)
448   Restriccion_G4HOLN:
449   G4HOLN[i] + MetasHOLN[i][1] - MetasHOLN[i][2] == TurnosDiasEspeciales["G4"]["
    HOLN"];
450 forall(i in RangoGrupo4)
451   Restriccion_G4HOLD:
452   G4HOLD[i] + MetasHOLD[i][1] - MetasHOLD[i][2] == TurnosDiasEspeciales["G4"]["
    HOLD"];
453 forall(i in RangoGrupo4)
454   Restriccion_G4WEN:
455   G4WEN[i] + MetasWEN[i][1] - MetasWEN[i][2] == TurnosDiasEspeciales["G4"]["WEN
    "];
456 forall(i in RangoGrupo4)
457   Restriccion_G4WED:
458   G4WED[i] + MetasWED[i][1] - MetasWED[i][2] == TurnosDiasEspeciales["G4"]["WED
    "];
459
460 //Médicos de G5:
461 forall(i in RangoGrupo5)
462   Restriccion_G5HOLN:

```

```

463 G5HOLN[i] + MetasHOLN[i][1] - MetasHOLN[i][2] == TurnosDiasEspeciales["G5"]["
    HOLN"];
464 forall(i in RangoGrupo5)
465 Restriccion_G5HOLD:
466 G5HOLD[i] + MetasHOLD[i][1] - MetasHOLD[i][2] == TurnosDiasEspeciales["G5"]["
    HOLD"];
467 forall(i in RangoGrupo5)
468 Restriccion_G5WEN:
469 G5WEN[i] + MetasWEN[i][1] - MetasWEN[i][2] == TurnosDiasEspeciales["G5"]["WEN
    "];
470 forall(i in RangoGrupo5)
471 Restriccion_G5WED:
472 G5WED[i] + MetasWED[i][1] - MetasWED[i][2] == TurnosDiasEspeciales["G5"]["WED
    "];
473
474 //Médicos de G6:
475 Restriccion_G6HOLN:
476 G6HOLN + MetasHOLN[42][1] - MetasHOLN[42][2] == TurnosDiasEspeciales["G6"]["
    HOLN"];
477 Restriccion_G6HOLD:
478 G6HOLD + MetasHOLD[42][1] - MetasHOLD[42][2] == TurnosDiasEspeciales["G6"]["
    HOLD"];
479 Restriccion_G6WEN:
480 G6WEN + MetasWEN[42][1] - MetasWEN[42][2] == TurnosDiasEspeciales["G6"]["WEN"];
481 Restriccion_G6WED:
482 G6WED + MetasWED[42][1] - MetasWED[42][2] == TurnosDiasEspeciales["G6"]["WED"];
483
484 //Médicos de G7:
485 Restriccion_G7HOLN:
486 G7HOLN + MetasHOLN[43][1] - MetasHOLN[43][2] == TurnosDiasEspeciales["G7"]["
    HOLN"];
487 Restriccion_G7HOLD:
488 G7HOLD + MetasHOLD[43][1] - MetasHOLD[43][2] == TurnosDiasEspeciales["G7"]["
    HOLD"];
489 Restriccion_G7WEN:
490 G7WEN + MetasWEN[43][1] - MetasWEN[43][2] == TurnosDiasEspeciales["G7"]["WEN"];
491 Restriccion_G7WED:
492 G7WED + MetasWED[43][1] - MetasWED[43][2] == TurnosDiasEspeciales["G7"]["WED"];
493
494 }

```

### Código 3: AsignacionMedicosExcel.dat

```

1 //Conectamos el libro de Excel indicando la ruta donde se encuentra.
2 SheetConnection sheetData("C:\\Users\\Usuario\\Documents\\TFG\\Programa
    asignacion medicos\\AsignacionMedicosSol.xlsx");
3
4 //Exportamos las matrices de variables de decisión.
5 //"TurnosS1", "TurnosS2",..., "MetasHOLD", "MetasHOLN" son los rangos definidos en el Excel.
6 AsignacionTurnosS1 to SheetWrite(sheetData,"TurnosS1");
7 AsignacionTurnosS2 to SheetWrite(sheetData,"TurnosS2");
8 AsignacionTurnosS3 to SheetWrite(sheetData,"TurnosS3");
9 AsignacionTurnosS4 to SheetWrite(sheetData,"TurnosS4");
10
11 MetasTurnosS1 to SheetWrite(sheetData,"MetasTurnosS1");
12 MetasTurnosS2 to SheetWrite(sheetData,"MetasTurnosS2");
13 MetasTurnosS3 to SheetWrite(sheetData,"MetasTurnosS3");
14 MetasTurnosS4 to SheetWrite(sheetData,"MetasTurnosS4");
15
16 MetasWED to SheetWrite(sheetData,"MetasWED");
17 MetasWEN to SheetWrite(sheetData,"MetasWEN");
18 MetasHOLD to SheetWrite(sheetData,"MetasHOLD");
19 MetasHOLN to SheetWrite(sheetData,"MetasHOLN");

```

## B.5. Resultados

Los resultados obtenidos en las variables metas se muestran en la Tabla B.9. Aunque el problema ha sido resuelto para todo el año 2020, debido a las grandes dimensiones de la solución, en la Tabla B.10 se muestra como ejemplo la planificación de los turnos en el SU obtenida para el mes de Enero. En esta asignación, los turnos se simbolizan con el número de horas que dura cada uno, de modo que un 8 indica que se ha asignado un turno de  $S_1$ , un 14 un turno de  $S_2$ , un 19.2 un turno de  $S_3$  y un 3 un turno de  $S_4$ . Finalmente, el 0 denota que no se ha asignado ningún turno.

	$d_{i,1}^-$	$d_{i,1}^+$	$d_{i,2}^-$	$d_{i,2}^+$	$d_{i,3}^-$	$d_{i,3}^+$	$d_{i,4}^-$	$d_{i,4}^+$	$c_{i,WED}^-$	$c_{i,WED}^+$	$c_{i,WEN}^-$	$c_{i,WEN}^+$	$c_{i,HOLD}^-$	$c_{i,HOLD}^+$	$c_{i,HOLN}^-$	$c_{i,HOLN}^+$
$M_1$	0,000	0,180	0,060	0,000	0,000	0,000	0,220	0,000	0,480	0,000	0,000	0,000	0,000	0,230	0,000	0,000
$M_2$	0,000	0,180	0,060	0,000	0,000	0,000	0,220	0,000	0,480	0,000	0,000	0,000	0,000	0,230	0,000	0,000
$M_3$	0,000	0,180	0,060	0,000	0,000	0,000	0,220	0,000	0,480	0,000	0,000	0,000	0,000	0,230	0,000	0,000
$M_4$	0,000	0,180	0,060	0,000	0,000	0,000	0,000	0,780	0,480	0,000	0,000	0,000	0,770	0,000	0,000	0,000
$M_5$	0,000	0,180	0,060	0,000	0,000	0,000	0,220	0,000	0,480	0,000	0,000	0,000	0,000	0,230	0,000	0,000
$M_6$	0,000	0,180	0,060	0,000	0,000	0,000	0,220	0,000	0,480	0,000	0,000	0,000	0,000	0,230	0,000	0,000
$M_7$	0,000	0,660	0,000	0,360	0,550	0,000	0,220	0,000	0,530	0,000	0,000	0,060	0,000	0,230	0,000	0,010
$M_8$	0,000	0,660	0,000	0,360	0,550	0,000	0,220	0,000	0,530	0,000	0,000	0,060	0,000	0,230	0,000	0,010
$M_9$	0,000	0,660	0,000	0,360	0,550	0,000	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{10}$	0,340	0,000	0,000	0,360	0,000	0,450	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{11}$	0,340	0,000	0,000	0,360	0,000	0,450	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{12}$	0,000	0,660	0,000	0,360	0,550	0,000	0,220	0,000	0,530	0,000	0,000	0,060	0,000	0,230	0,000	0,010
$M_{13}$	0,340	0,000	0,000	0,360	0,000	0,450	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{14}$	0,000	0,660	0,000	0,360	0,550	0,000	0,000	0,780	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{15}$	0,000	0,660	0,000	0,360	0,550	0,000	0,220	0,000	0,530	0,000	0,000	0,060	0,000	0,230	0,000	0,010
$M_{16}$	0,340	0,000	0,000	0,360	0,000	0,450	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{17}$	0,340	0,000	0,000	0,360	0,000	0,450	0,000	0,780	0,530	0,000	0,000	0,060	0,770	0,000	0,000	0,010
$M_{18}$	0,340	0,000	0,000	0,360	0,550	0,000	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{19}$	0,340	0,000	0,000	0,360	0,000	0,450	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{20}$	0,340	0,000	0,000	0,360	0,000	0,450	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{21}$	0,000	0,660	0,000	0,360	0,550	0,000	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{22}$	0,340	0,000	0,000	0,360	0,000	0,450	0,220	0,000	0,530	0,000	0,000	0,060	0,000	0,230	0,000	0,010
$M_{23}$	0,340	0,000	0,000	0,360	0,000	0,450	0,000	0,780	0,530	0,000	0,000	0,060	0,000	0,230	0,000	0,010
$M_{24}$	0,000	0,660	0,640	0,000	0,000	0,450	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{25}$	0,340	0,000	0,000	0,360	0,550	0,000	0,000	0,780	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{26}$	0,000	0,660	0,000	0,360	0,550	0,000	0,220	0,000	0,530	0,000	0,000	0,060	0,000	0,230	0,000	0,010
$M_{27}$	0,340	0,000	0,000	0,360	0,550	0,000	0,000	0,780	0,530	0,000	0,000	0,060	0,000	0,230	0,000	0,010
$M_{28}$	0,000	0,660	0,000	0,360	0,550	0,000	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{29}$	0,340	0,000	0,000	0,360	0,000	0,450	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{30}$	0,340	0,000	0,000	0,360	0,000	0,450	0,220	0,000	0,000	0,470	0,000	0,060	0,000	0,230	0,000	0,010
$M_{31}$	0,180	0,000	0,480	0,000	0,000	0,450	0,220	0,000	0,000	0,290	0,000	0,240	0,730	0,000	0,040	0,000
$M_{32}$	0,180	0,000	0,480	0,000	0,000	0,450	0,220	0,000	0,000	0,290	0,000	0,240	0,730	0,000	0,040	0,000
$M_{33}$	0,180	0,000	0,480	0,000	0,000	0,450	0,220	0,000	0,000	0,290	0,000	0,240	0,730	0,000	0,040	0,000
$M_{34}$	0,260	0,000	0,560	0,000	0,000	0,450	0,220	0,000	0,120	0,000	0,350	0,000	0,250	0,000	0,000	0,480
$M_{35}$	0,260	0,000	0,560	0,000	0,550	0,000	0,220	0,000	0,120	0,000	0,350	0,000	0,250	0,000	0,000	0,480
$M_{36}$	0,260	0,000	0,560	0,000	0,000	0,450	0,000	0,780	0,120	0,000	0,350	0,000	0,250	0,000	0,520	0,000
$M_{37}$	0,260	0,000	0,560	0,000	0,550	0,000	0,220	0,000	0,120	0,000	0,350	0,000	0,250	0,000	0,000	0,480
$M_{38}$	0,260	0,000	0,560	0,000	0,550	0,000	0,000	0,780	0,120	0,000	0,350	0,000	0,250	0,000	0,000	0,480
$M_{39}$	0,260	0,000	0,560	0,000	0,550	0,000	0,220	0,000	0,120	0,000	0,350	0,000	0,250	0,000	0,520	0,000
$M_{40}$	0,290	0,000	0,630	0,000	0,000	0,450	0,000	0,780	0,000	0,170	0,000	0,350	0,510	0,000	0,250	0,000
$M_{41}$	0,290	0,000	0,630	0,000	0,000	0,450	0,220	0,000	0,000	0,170	0,000	0,350	0,510	0,000	0,250	0,000
$M_{42}$	0,200	0,000	0,480	0,000	0,000	0,110	0,000	0,000	0,000	0,420	0,630	0,000	0,320	0,000	0,480	0,000
$M_{43}$	0,120	0,000	0,450	0,000	0,000	0,110	0,000	0,000	0,170	0,000	0,040	0,000	0,000	0,200	0,010	0,000

Tabla B.9: Valores resultado de las variables metas  $d_{i,j}^+$ ,  $d_{i,j}^-$ ,  $c_{i,k}^+$ ,  $c_{i,k}^-$ , para todo  $i = 1, \dots, N$ ;  $j = 1, \dots, L$ ;  $k \in K$ .

